

Lecture Notes in Computer Science

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

2465

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Tokyo

Hiroshi Arisawa Yahiko Kambayashi
Vijay Kumar Heinrich C. Mayr
Ingrid Hunt (Eds.)

Conceptual Modeling for New Information Systems Technologies

ER 2001 Workshops
HUMACS, DASWIS, ECOMO, and DAMA
Yokohama, Japan, November 27-30, 2001
Revised Papers



Springer

Volume Editors

Hiroshi Arisawa

Yokohama National University

Graduate school of environment and information sciences

79-7, Tokiwadai, Hodogaya-ku, yokohama 240-8501, Japan

E-mail: arisawa@ynu.ac.jp

Yahiko Kambayashi

Kyoto University

Department of Social Informatics, Graduate School of Informatics

Yoshida, Sakyo, Kyoto 606-8501, Japan

E-mail: yahiko@i.kyoto-u.ac.jp

Vijay Kumar

University of Missouri-Kansas City, SICE Computer Networking

5100 Rockhill Road, Kansas City, MO 64110, USA

E-mail: kumarv@umkc.edu

Heinrich C. Mayr

University of Klagenfurt

Universitätsstraße 65-67, 9020 Klagenfurt, Austria

E-mail: mayr@ifit.uni-klu.ac.at

Ingrid Hunt

VP Industry Services DAMA International

PO Box 5786, Bellevue, WA 98006-5786, USA

E-mail: vp_industry_services@dama.org

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Conceptual modeling for new information systems technologies : revised papers /

ER 2001 workshops HUMACS ..., Yokohama, Japan, November 27 - 30, 2002.

Hiroshi Arisawa ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ;

Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002

(Lecture notes in computer science ; Vol. 2465)

ISBN 3-540-44122-0

CR Subject Classification (1998): H.2, H.3-5, C.2.4-5, J.1

ISSN 0302-9743

ISBN 3-540-44122-0 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York,
a member of BertelsmannSpringer Science+Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2002

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna e.K.
Printed on acid-free paper SPIN: 10871259 06/3142 5 4 3 2 1 0

Preface

The objective of the workshops associated with ER 2001, the 20th International Conference on Conceptual Modeling, was to give participants the opportunity to present and discuss emerging hot topics, thus adding new perspectives to conceptual modeling. This, the 20th ER conference, the first of the 21st century, was also the first one in Japan. The conference was held on November 27-30, 2001 at Yokohama National University with 192 participants from 31 countries. ER 2001 encompasses the entire spectrum of conceptual modeling, from theoretical aspects to implementations, including fundamentals, applications, and software engineering. In particular, ER 2001 emphasized e-business and reengineering. To meet this objective, we selected the following four topics and planned four international workshops:

- International Workshop on Conceptual Modeling of Human/Organizational/Social Aspects of Manufacturing Activities (HUMACS 2001)

Manufacturing enterprises have to confront a host of demands. The competitive climate, enhanced by communication and knowledge sharing, will require increasingly rapid responses to market forces. Customer demands for higher quality, better services, and lower cost will force manufacturers to reach new levels of flexibility and adaptability. Sophisticated customers will demand products customized to meet their needs. Industries have so far sought to cope with these challenges primarily through advances in traditional capital by installing more powerful hardware and software technology. Attention to the role of humans combined with organizational and social schemes in manufacturing has only been marginal.

The workshop HUMACS 2001 aimed to challenge the relevance of this last point. The basis for competition will be creativity and innovation in all aspects of manufacturing enterprises. This will emphasize on the one hand the importance of knowledge and knowledge processing in organizations. It will necessarily shed light on the need for establishing such working environments as will give full play to the abilities of individuals by ensuring human dignities that have often been damaged by mere pursuit of efficiency in modernized manufacturing systems. The objective of this workshop was to exchange information on recent advances in information modeling, simulation, and database design centered on human / organizational / social aspects of manufacturing activities as a challenge to address issues on human factors in relation to working environments and organizations encountered in the design and operation of manufacturing systems.

- International Workshop on Data Semantics in Web Information Systems: DASWIS 2001

Information systems are rapidly being influenced by web technologies and are driven by their growth and proliferation to create next generation web information systems. A major issue that crops up is the conceptualization of the symbiotic relationship between the old economy based client server information systems and the new economy driven web information systems. Most of this symbiotic relationship

is hidden in the data transformation and routing that takes place in the middle-ware connecting the databases used for transaction processing in the old economy to (web) data warehouses that are also needed to support on-line analytical processing in the new economy.

The processing of data across these disparate systems is hidden in the loosely coupled applications that support the business processes. This motivates the need to extract the semantics of the underlying data and its processing, and make it explicit. Further, the new economy is triggering the manifestation of applied technologies such as, workflow systems, multiagent-based information systems, and home automation systems. Moreover, agent technologies are being used to provide information management support by facilitating cooperative information processing across information systems of old and new economies.

The focus of this workshop is on the modeling of data semantics for facilitating data dissemination and use, secure access, and modification and transfer of data, and conceptual modeling and design of next generation web information systems.

– 2nd International Workshop on Conceptual Modeling Approaches for e-Business: eCOMO 2001

The network economy comes with new challenges for information systems developers and users. They will have to supply and to adopt web-based services which will go far beyond the actual solutions for B2B (business to business) and B2C (business to commerce). Conceptual modeling plays an important role within this context. For example, standardized process models and methodological means are needed for the coupling and de-coupling of (heterogeneous) information systems, as is needed for an enhanced formation of virtual enterprises and dynamic enterprise networks, or for the transition from conventional business processes to globalized e-Business. eCOMO 2001 was the continuation of a highly successful first workshop which was held during ER 2000 in Salt Lake City. Experts from practice and academia were cordially invited to exchange and discuss actual developments, methods and tools, as well as their experiences in using these. Thus, the workshop contributed to intensifying research within that domain and to determining directions for future research.

– DAMA 2001 International Workshop: Global Data Modeling in the New Millennium

This was an exciting opportunity for DAMA International to gain wider international visibility and to exchange experiences with an international audience.

Within the topic of Data Modeling lies a vast array of real-life global factors based on approach, method, and design interpretation. From the theoretical foundations to database implementation strategies, there are tangible and meaningful lessons to be learned in following different methods to reach a global view. Knowing which method to use based on the needs of the business environment and detailed specifications will affect the globalization of the requirements. Whether modeled for a business or personal environment, all methods entail life long learning experiences for the international client, user, and implementers.

The DAMA International sponsored workshop, addresses topics exploring the global impact of the Data Model, and its cause and affect within various industry expe-

periences. How do we really embody conceptualization and/or theory and share this interpretation globally? What happens when we use one method versus another? What would have been the cause and affect if we had skipped a particular process path, or used another method or approach? What would we do differently to handle global standards? How did we make the data model real and tangible to the global user, developer, and/or consumer?

Emphasis is placed on international speakers from different countries that can provide a wide variety of experiences with internationalization of data modeling.

Following the philosophy of the ER workshops, the call for papers and selection of contributions were done very carefully by each workshop program committee in order to guarantee an excellent workshop program. This book contains selected workshop papers, which were presented in one of the workshop sessions and revised by the authors following the discussions held during the conference. (However, we were only able to include one paper from the DAMA workshop, as many of the presentations were given without a technical paper.) We are deeply indebted to the authors and to the members of the program committee, whose work resulted in this outstanding program. We acknowledge the hard work of the many individuals who made these workshops a great success.

We would like to express special appreciation to Mr. Yoshioki Ishii, the ER 2001 Conference Chair.

Juli 2002

Hiroshi Arisawa
Yahiko Kambayashi
Vijay Kumar
Heinrich C. Mayr
Ingrid Hunt

HUMACS 2001 Workshop Organization

General Chair

Hiroshi Arisawa, Yokohama National University, Japan

Program Co-chairs

Eiji Arai, Osaka University, Japan

Jan Goossenaersts, Eindhoven University of Technology, The Netherlands

Area Coordinators

Noriaki Kurosu, Toyota Motor Corporation, Japan

Takashi Tomii, Yokohama National University, Japan

Shunji Yamada, Yamatake Industrial Systems Co., Japan

Program Committee Members

Hiroshi Arisawa (Yokohama National University, Japan)

Peter Bertok (RMIT University, Australia)

Gudela Grote (ETH, Switzerland)

Peter Groumpos (University of Patras, Greece)

Kiyoshi Itoh (Sophia University, Japan)

Satoshi Kumagai (Yamatake Corporation, Japan)

Daniel Lebranc (Ecole Polytechnique Montreal, Canada)

John J. Mills (University of Texas, USA)

Narinder Nayar (Delmia Corporation, USA)

Kageyu Noro (Waseda University, Japan)

Peter Orban (IAW/RWTH, Germany)

Jorma Saari (FIOH, Finland)

Sadananda (Asia Institute of Technology, Thailand)

Johan Stahre (Chalmers University, Sweden)

Frans van Eijnatten (Eindhoven University of Technology, The Netherlands)

Peter Vink (TNO, The Netherlands)

DASWIS 2001 Workshop Organization

General Chair

Vijay Kumar, University of Missouri-Kansas City, USA

Program Co-chairs

Kamal Karlapalem, International Institute of Information Technology, India
Mukesh Mohania, Western Michigan University, USA

Area Coordinators:

Albert Burger, Heriot-Watt University, UK
Sanjay Madria, University of Missouri at Rolla, USA

Program Committee Members

Shan, Ming-Chien (HP Labs., California, USA)
Sunil Prabhakaran (Purdue University, USA)
Maggie Dunham (Southern Methodist University, USA)
Panos K. Chrysanthis (University of Pittsburgh, USA)
Sang Son (University of Virginia, USA)
Masaru Kitsuregawa (University of Tokyo, Japan)
Stefano Rizzi (University of Bologna, Italy)
Elke Rundensteiner (Worcester Polytechnic Institute, USA)
Bharat Bhargava (Purdue University, USA)
Sham Navathe (Georgia Tech, USA)
Chris Rainsford (DSTO, Australia)
Sourav Bhowmick (Nanyang Technological University, Singapore)
A Min Tjoa (Technical University of Vienna, Austria)
P Krishna Reddy (University of Tokyo, Japan)
Philippe Rochat (EPFL, Switzerland)
Li Yang (Western Michigan University)
P. K. Chande (MISEM Software Institute, Pune, India)
Peter I. Scheuermann (Northwestern University, USA)
S.R. Subramanya (University of Missouri-Rolla, USA)
Tan Kian Lee (National University of Singapore, Singapore)
Georg Lausen (University of Freiburg, Germany)
Wee Keong Ng (NTU, Singapore)
Giuseppe Psaila (University of Bergamo, Italy)
Vijay Atluri (Rutgers University, USA)
Tiziana Catarci (Università degli Studi di Roma "La Sapienza", Italy)
Dickson Chiu (Dickson Computer Systems, HKSAR, China)
Qing Li (City University of Hong Kong, HKSAR, China)
Fred Lochovsky (HKUST China)
Barbara Pernici (Politecnico di Milano, Italy)
Guenther Pernul (University of Essen, Germany)
Eric Yu (University of Toronto, Canada)
Ghassan Al-Qaimari (RMIT, Australia)
Nimal Jayaratna (Curtin University of Technology, Australia)

Lachlan MacKinnon (Heriot-Watt University, UK)
Moirra C. Norrie (ETH Zentrum, Switzerland)
Janakiram (IIT, Madras, India)
Ling Tok Wang (NUS, Singapore)
Gerd Wagner (Eindhoven University of Technology, The Netherlands)
Keith Jeffery (Rutherford Appleton Laboratory, UK)
Carole Goble (University of Manchester, UK)
Gi-Chul Yang (Mokpo National University, Korea)
Umesh Dayal (HP, USA)
Ladjel Bellatreche (Grenoble I University, France)
Takao Miura (Hosei University, Japan)
Rajeev Agrawal (Western Michigan University, USA)

eCOMO 2001 Workshop Organization

General Chair

Heinrich C. Mayr, University of Klagenfurt, Austria

Program Committee Members

Boldur Barbat (Lucian Blaga University, Sibiu, Romania)

Jaap Gordijn (Vrije Universiteit Amsterdam, The Netherlands)

Nicola Guarino (National Research Council, Padova, Italy)

Jozsef Györkös

(Faculty of Electrical Engineering and Computer Science, Slovenija)

Bill Karakostas (UMIST Manchester, UK)

Roland Kaschek (UBS Zurich, Switzerland)

Jacques Kouloumdjian (INSA Lyon, France)

Stephen Liddle (Brigham Young University, USA)

Norbert H. Mikula (DataChannel, Seattle, USA)

Oscar Pastor (University of Valencia, Spain)

Klaus-Dieter Schewe (Massey University, New Zealand)

Daniel Schwabe (PUC-Rio, Brazil)

Il-Yeol Song (Drexel University, Philadelphia, USA)

Rudi Studer (University of Karlsruhe, Germany)

Bernhard Thalheim (BTU Cottbus, Germany)

Christopher Welty (Vassar College/IBM, Poughkeepsie, USA)

Carson Woo (UBC Vancouver, Canada)

DAMA 2001 Workshop Organization

General Chair Liaison

Michael Brackett, DAMA President

Ingrid Hunt, DAMA VP, Industry Services

Onsite Co-chair Representative

Peter Aiken, Virginia Commonwealth University, USA

Table of Contents

HUMACS 2001

Participation: The Key to Intelligent Manufacturing Improvement	1
<i>Peter Vink, Frans M. van Eijnatten</i>	
Challenges in Dealing with Human Factors Issues in Manufacturing Activities	10
<i>Shunji Yamada</i>	
Extracting E-R Models from Collaboration Analysis Methods, MCM, and CLM	30
<i>Manabu Kamimura, Naoyuki Kato, Ryo Kawabata, Satoshi Kumagai, Kiyoshi Itoh</i>	
Living Manufacturing Systems with Living Organizations	44
<i>Noriaki Kurosu, Shunji Yamada</i>	
A Study on Human-Centric Real-Time Scheduling for PWB Assembly Line	57
<i>Machiko Chikano, Yoshitaka Tomita, Yasuhiko Hiraide, Eiji Arai</i>	
The E/S Tool IT-Support for Ergonomic and Sociotechnical System Design	67
<i>Martin Van De Bovenkamp, Ruben Jongkind, Gu Van Rhijn, Frans M. Van Eijnatten, Gudela Grote, Jouni Lehtelä, Timo Leskinen, Peter Vink, Scott Little, Toni Wüfler</i>	
Construction of Virtual Working Environment and Evaluation of the Workers	81
<i>Kageyu Noro, Ryohei Tanaka</i>	
Human Models and Data in the Ubiquitous Information Infrastructure	91
<i>Frank Berkers, Jan Goossenaerts, Dieter Hammer, Hans Wortmann</i>	
Motion Simulation of the Human Workers for the Integrated Computer-Aided Manufacturing Process Simulation Based on Info-Ergonomics Concept	105
<i>Sayaka Imai, Takashi Tomii, Hiroshi Arisawa</i>	
Human-Body Motion Simulation Using Bone-Based Human Model and Construction of Motion Database	115
<i>Hiroshi Arisawa, Takako Sato, Takashi Tomii</i>	
Ontological Commitment for Participative Simulation	127
<i>Jan Goossenaerts, Christine Pelletier</i>	

Dynamic Management Architecture for Human Oriented Production System	141
<i>Keiichi Shirase, Hidefumi Wakamatsu, Akira Tsumaya, Eiji Arai</i>	

DASWIS 2001

GeoCosm: A Semantics-Based Approach for Information Integration of Geospatial Data	152
<i>Sudha Ram, Vijay Khatri, Limin Zhang, Daniel Dajun Zeng</i>	
Imposing Modeling Rules on Industrial Applications through Meta-modeling	166
<i>Peter Fröhlich, Zaijun Hu, Manfred Schoelzke</i>	
Modelling Ubiquitous Web Applications – The WUML Approach	183
<i>Gerti Kappel, B. Pröll, Werner Retschitzegger, Wieland Schwinger</i>	
Structuring Web Sites Using Audience Class Hierarchies	198
<i>Sven Casteleyn, Olga De Troyer</i>	
On the Automatic Extraction of Data from the Hidden Web.....	212
<i>Stephen W. Liddle, Sai Ho Yau, David W. Embley</i>	
MIDAS/BD: A Methodological Framework for Web Database Design	227
<i>E. Marcos, P. Cáceres, B. Vela, J.M. Cavero</i>	
Translating XQuery into XSLT	239
<i>Stephan Lechner, Günter Preuner, Michael Schrefl</i>	
Web Site Evaluation: Methodology and Case Study	253
<i>Paolo Atzeni, Paolo Merialdo, Giuseppe Sindoni</i>	
Automatic Web Information Extraction in the ROADRUNNER System	264
<i>Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo</i>	
Querying Relational Databases without Explicit Joins	278
<i>Ramon Lawrence, Ken Barker</i>	
NF-SS: A Normal Form for Semistructured Schema	292
<i>Xiaoying Wu, Tok Wang Ling, Sin Yeung Lee, Mong Li Lee, Gillian Dobbie</i>	
A Formal Analysis of the Lightweight Directory Access Protocol	306
<i>Fang Wei, Georg Lausen</i>	
An XML Document Retrieval System Supporting Structure- and Content-Based Queries	320
<i>Jae-Woo Chang</i>	

Extraction of Partial XML Documents Using IR-Based Structure and Contents Analysis	334
<i>Kenji Hatano, Hiroko Kinutani, Masatoshi Yoshikawa, Shunsuke Uemura</i>	
XDoC-WFMS: A Framework for Document Centric Workflow Management System	348
<i>Rupa Krishnan, Lalitha Munaga, Kamalakar Karlapalem</i>	
eCOMO 2001	
Active XML Schemas	363
<i>Michael Schrefl, Martin Bernauer</i>	
Behavior Abstraction in Semantic B2B Integration	377
<i>Christoph Bussler</i>	
OIL Ontologies for Collaborative Task Performance in Coalitions of Self-Interested Actors	390
<i>Vadim Ermolayev, Natalya Keberle, Vyachyslav Tolok</i>	
A Multi-perspective Methodology for Modelling Inter-enterprise Business Processes	403
<i>Kuldar Taveter, Gerd Wagner</i>	
Process Patterns to Generate E-commerce Systems	417
<i>Prasad Jayaweera, Paul Johannesson, Petia Wohed</i>	
Formalising Feasibility and Correctness of Distributed Business Processes	432
<i>Tomasz Janowski, Adegboyega Ojo</i>	
Modeling Products for Versatile E-commerce Platforms – Essential Requirements and Generic Design Alternatives	444
<i>Ulrich Frank</i>	
Seamless Personalization of E-commerce Applications	457
<i>Juan Cappi, Gustavo Rossi, Andrés Fortier, Daniel Schwabe</i>	
Discovery of User Preference through Genetic Algorithm and Bayesian Categorization for Recommendation	471
<i>SuJeong Ko, JungHyun Lee</i>	
DAMA 2001	
Using the Quantum Data Model to Develop Shareable Definitions	485
<i>Harry Ellis</i>	
Author Index	499

Participation: The Key to Intelligent Manufacturing Improvement

Peter Vink¹ and Frans M. van Eijnatten²

¹TNO Arbeid, P.O. Box 718,
2130 AS Hoofddorp, The Netherlands,
P.Vink@arbeid.tno.nl

²Eindhoven University of Technology, Faculty of Technology Management, Pav.U10-T&A,
P.O. Box 513,
5600 MB Eindhoven, The Netherlands,
F.M.v.Eijnatten@tm.tue.nl

Abstract. This paper describes the background and objectives of the IST project “Organizational Aspects of Human-Machine Coexisting Systems” (HUMACS), that develops and pilot-demonstrates a Participative Simulation environment for Integral (i.e., logistics, technology and human factors) Manufacturing enterprise renewal (PSIM). In the short run, HUMACS/PSIM aims to address the key issues that have to be resolved before a virtual factory can be implemented, and is focused on intelligent manufacturing to assist human creativity and the application of ICT technology. The long-term goal is to show simulated assembly lines in a software environment, to be used in assembly operations to enable a participative improvement process involving specialized staff, management and production personnel. This paper illustrates both the principle and developed prototype of the PSIM tool, including its STEAM procedure.

1 Introduction

Today’s manufacturing enterprises have to optimize their production in a highly competitive and global market place, at an ever increasing rate. In order to survive, they have to confront a multitude of stakeholders’ demands, and have to improve the process of product creation, in order to turn an innovative prototype into a manufacturable product, at a much higher speed than ever before. At the same time, the resulting designs for new products should fit manufacturing systems that can cost-effectively produce them. These trends accentuate the significance of knowledge and knowledge processing in organizations.

Most modern organizations are on their way to become knowledge-based enterprises. Prominent researchers in knowledge management [1] [2] advocate, that knowledge should be seen as the resource for competitive advantage in current and future competition. They emphasize the management of knowledge at the organizational level, usually called “competencies”.

2 Participation and Simulation

In organizational design and management ever more attention is paid to successful improvement processes, enabled by participation [6]. Participation is defined as a process which allows employees to exert some influence over their work and the conditions under which they work [7]. According to these authors, competence (capability) is "both a requirement for and a consequence of participation" [7], p. 45. It is a requirement because participation needs a minimum level of skills in order to be effective. It is a consequence because participation enhances the skills levels of those involved. Participation as a process has advantageous results for both the individual—in terms of capability and job satisfaction, and for the organization—in terms of core competence, increased efficiency and effectiveness.

Some of the benefits of participation are recognized in its contribution to a smooth mutual communication between management and employees. ICT can support this communication by providing highly visual representations of abstract processes, that conduce as a common basis for discussions and suggestions. Participation may be considered in the development, the implementation and the application of an improvement project. To boost participation, it is recommended to allow the employees to establish cross-departmental task teams, which deal with improvement issues.

The European Foundation for the Improvement of Living and Working Conditions [8] reports that direct participation in organizations most often leads to some quality improvements (90% of the cases), to some reduction of throughput times (60% of the cases), and to some reduction of costs (60% of the cases).

ICT can support communication by providing visual representations of abstract processes, that contribute to a common basis for discussions and suggestions. In this respect, simulation is the construction and use of a computer-based representation, or model, of some part of the real world as a substitute vehicle for experiment and behavior prediction. It offers an attractive opportunity for engineers, planners, managers and teams to try out ideas in advance of commitment to a course of action [9].

Participation could help improving the work of the manufacturing work force. A powerful integrated digital environment that would bring to life a virtual copy of the actual manufacturing system thus represents an interesting facility [10] [11] [12]. It would enable profound analysis of possible interventions in the real manufacturing system, and ensure much more efficient improvement efforts [13].

3 Participative Simulation

In participative simulation, workers exert direct influence over the product and process designs by bringing in their tacit knowledge, to combine it with expert knowledge, and to put the blend of both insights to the test. When these experimenting and problem solving activities be supported by an attractive ICT interface, the resulting continuous improvement process may become even more

intrinsically motivating for the work force [14]. Besides, it also contributes to the competitive advantage of the organization.

Unfortunately, the total number of users of simulation tools is still low. In so far (simulation) tools are used to reflect on possible interventions, they are often stand-alone and do not support an integrated perspective on possible changes to reality. One of the really big problems is to generate a common description, e.g. a future work station or situation on the shop floor in the existing plant, and let people with various experiences participate in the analysis. In addition, the tools are not connected to the real world, and therefore, often reflect a state of the business, that is outdated.

In participative simulation, the tools should not only produce images (mere descriptions) of all sorts of designs, but also should be able to compare their respective qualities (evaluations) and suggest improvements (reflections, and regulative actions). In order to accomplish that, the tools should be upgraded to expert bases. Although the technical aim in advanced intellectual manufacturing systems more than once is to accomplish a more predictable work system, experiences from the social sciences indicate, that especially when humans are concerned, absolute norms and solid predictability are not feasible, and absolute centralized control is an utopia.

In PSIM, the participative simulation combines both holistic and analytic thinking, at different levels of aggregation, at the same time. At the highest level, an image of the whole is created. At the lowest level, variables and interrelationships are shown. It is the aim of the supporting ICT architecture, that all sorts of simulation tools can be easily plugged in, in order to support the process of dialogue. The idea is to both tap explicit and implicit knowledge, at the same time.

4 Intellectual Capital Management in Assembly Operations

Although the idea of participative simulation is not new, the potential of this method in organizations was rather restricted, for a long time. It is the development of modern ICT technologies that expands the participative simulation potentials, with an order of magnitude. The local knowledge of workers, locked in their traditions and work habits, may be successfully tapped and communicated by using ICT-supported participative simulation.

Traditionally, the assembly industry has shown a tendency to try to embed such knowledge in the material components of the assembly line and its control system. The decreasing level of specificity in the material components of the current manufacturing system that are based on openness and modularity, implies that the competitive advantage of the manufacturing system as a whole has to be found elsewhere: The human operator and his work methods come to stand out more prominently. It is known, that these work methods develop on the basis of complex and unique organizational culture and strategies, and “know-how”, and consequently, are not easily imitated [14] [15].

The centrality of the human factor calls for “Intellectual Capital Management”. Although Intellectual Capital Management has received a lot of attention in professional service organizations – and evolved there into a hype – it has been

almost completely neglected in assembly. If a company wants to make efficient use of knowledge and intends to cause the knowledge, skills and experience of its employees to become more effective with respect to achieving organizational goals, the two perspectives on Intellectual Capital Management, organizational and individual competencies, should be aligned. The two perspectives become complementary to each other. Unfortunately, theory does not provide much guidance on how to accomplish that.

Typically, most researchers on core competencies are not explicitly stating their level of analysis: They do not clearly distinguish organizational from individual competencies. Core competencies are discussed as collection and integration of skills and technology of a company as a whole (across diverse business units). Individual employees are seen as the “skills carriers” that embody the competencies [16] [17]. These theorists recognize that in practice, a mechanism for allocating skills, is seriously lacking. Hamel and Prahalad write: “We find it ironic that top management devotes so much attention to the capital budgeting process, yet typically has no comparable mechanism for allocating the human skills that embody core competencies” [16], p. 89. But they do not discuss any method or approach how to fill in the role of individual capabilities, with respect to strategic objectives and competencies of an organization. A similar conclusion can be drawn from literature on individual capabilities: A clear connection with organizational goals and core competencies is lacking.

To take a step forward, we propose an approach based on the idea that the power of knowledge is not so much leveraged by having knowledge. It is far more important to know, how to allocate knowledge for productive use [1]. Nonaka and Takeuchi discuss the role of an organization in allocating knowledge: “The organization supports creative individuals or provides con-texts for them to create knowledge. Organizational knowledge creation, therefore, should be understood as a process that ‘organizationally’ amplifies the knowledge created by individuals and crystallizes it as a part of the knowledge network of the organization” [4], p. 59. The successful execution of this organizational activity can be regarded as a core competence.

At this point we can refer to the point made in the introduction. To really become a competitive strength, the work methods should reflect all manufacturing expertise that is available in an organization, not only the insights of a privileged process engineering elite. In the design of new systems, or reconfiguring of existing ones, interdisciplinary participative reflection should be encouraged and supported, to influence the manufacturing organization primarily as a knowledge-processing entity. In order to accomplish this goal, strategic action through investments in a proper environment is needed. Computerized facilities for “participative simulation” could be instrumental in this matter.

5 Objectives, Significance, and Benefits of PSIM

PSIM aims for the development of a simulation environment for use in assembly operations and to advance integral renewal in a competitive, changing environment by supporting continuous improvement processes. In this project simulated assembly

lines are developed and pilot-demonstrated in a software environment to be used in an improvement process involving specialized staff, management and production personnel. By the end of the project a structure for a software environment as well as a process of implementation is developed that is proven to be operational in 3 EU-pilot sites and is studied with other HUMACS partners including other potential PSIM-users in the EU, Switzerland, Japan, and the USA.

At Roberine, a company manufacturing mowing machines, applying the multi-disciplinary participative assembly simulation resulted in 15% efficiency improvement and 25% better working conditions (ROI 20%, see [18]). However, this process did not include the PSIM software, so it took much time to visualize the different forms of production lay-out. The significance of the PSIM project is, that this process can be shortened and made of a much higher quality. We expect a 15% efficiency improvement at the three pilot sites and 20% better work satisfaction due to better working conditions in about two years.

In order to test the basic ideas, the PSIM project will actively engage into a reality check, using several industrial test sites in Europe, Japan and the USA. To demonstrate the concept, PSIM will concentrate on expertise from the domains of Socio-Technical Systems Design and Ergonomics. One key design methodology is Socio-Technical Systems Design (STSD), which is concerned with the optimization and integration of the human factor in manufacturing systems, predominantly at the work group, departmental and organizational levels. It aims at improving the quality of work and organization, simultaneously, through adaptation or fundamental redesign of contents and composition of technology and human tasks [19]. The Dutch STSD variant of Integral Organizational Renewal (IOR) offers dedicated design concepts, methods and strategies. These can be used for diagnosing and improving existing production structures in order to make optimal use of the human factor, while at the same time enabling a multitude of design objectives (i.e. innovation, flexibility, controllability and quality of work). STSD can successfully support ICT-driven simulation of organizational renewal in a development activity game environment. Within the socio-technical framework, also a method was developed that specifically addresses the issue of allocating tasks between humans and technology, i.e. defining the degree of automation. Key to this method are design criteria on the level of work system, individual task, and human-machine interaction, which can be used also in system modeling and simulation [20] [21].

While the focus of the socio-technical framework is the human-human-technology interaction, the more specific aspects of fitting tasks and technology to human operators is dealt with by an ergonomic approach concerned with optimizing the tasks, technical systems and work stations in order to improve human performance and to reduce mental and physical workload. Data from the European Foundation for the Improvement of Living and Working Conditions [8] indicate that a rise in 'time pressure' has taken place through-out Europe. Approximately 30% of the workers in the European Union are involved in painful and tiring postures for more than half of their working day and 40% of the workers are exposed to short repetitive tasks, which often lead to reduced quality, productivity, complaints or even sick leave. A recent survey reports on the work-relatedness of drop out from work due to psychological dysfunctioning. Some important aspects in the reduction of workload are the good fit between task and personality, possibilities to develop and regulate your own work.

Therefore an important function in PSIM is envisioned that will warn users when unacceptable workload for humans and teams is anticipated in a particular work system design. Users of PSIM will be warned for physical and mental hazards in designs of a workflow or workstation.

6 First Results from a Test

The procedure as well as the ergo- and STSD-tool were tested in three companies in the end of 2002. In all three test sites it was shown that the procedure was an essential part. The steps of analysing the current situation en discussing ideas for improvement with a group of engineers, operators, management and designers were both evaluated positive. The companies evaluating the procedure mentioned that a facilitator is very much needed. The experts on ergonomics and sociotechnique were essential in supporting to follow the procedure and in explaining the backgrounds. Also, the visualisation with the video was evaluated positive. Regarding both tools there were differences between companies. The companies that were not very active in applying ergonomics evaluated the ergo tool more positive then the others. Application of the mental workload module in the ergo tool and the sociotechnical tool resulted in most new improvement ideas and where evaluated very positive. Other parts were nice, but it was the question whether they will replace existing checklists, methods or software that are already used in companies. It was mentioned that application of the STSD-tool was time consuming. The ICT part of PSIM was difficult to test, because loading and building the systems would take too much time of the companies. Nevertheless, it was stated that an ontology that enables communication between different software packages is welcome as well as having an abstract of the ERP to be able to test effects. The navigator, software that navigates the end-user through a system, will be tested later.

7 Conclusions and Recommendations

Since the PSIM project only started in March 2000, any results are intermediate. However, it is to be expected that the project will produce a breakthrough in both participative simulation method and ICT architecture, including the ontology. It is anticipated, that the ICT architecture will enable other knowledge domains to be integrated in the PSIM tool as modules quite easily. A potential candidate for inclusion is the “Design of Workspace II” decision-making model that resulted from the Brite-Euram III, Workspace II Thematic Network BET2-516, 4th Framework Program [22], that will add decision making about facility management to the participative simulation environment.

The lessons learned thus far concentrate on the topics of interdisciplinary preparation and communication. It appeared a necessity to visit the test sites with a full multi-disciplinary team, in order to research the requirements, in order to develop and test the tool appropriately. During the development of the ontology, major differences in concepts and methodologies among the experts came to the fore. The

readiness to take enough time to dialogue about these issues extensively, proved a prerequisite to solve these issues. It offered a basis for a successful completion of the PSIM project.

As to main barriers for adoption, a problem could be the overall attractiveness of the simulation tool for the end users, or the modest level of penetration of computers in assembly operations. Also, the generality of the tool may be questioned, in specific assembly environments. Cultural differences may influence the potential usability of the tool in either Europe, Japan or the USA.

Note

The PSIM/HUMACS project was presented earlier at the ISATA conference [23], the IMS Workshop in Leuven [24], and the IST conference in Helsinki [14] [25].

Acknowledgements. This paper has described the background and objectives of the PSIM project which is part of the HUMACS project within the international IMS research program.

The members of the PSIM consortium are: Dr. P. Vink, Dr. G. van Rhijn, Dr. E. Cox-Woudstra, and Mrs. H. Knijnenburg (TNO Arbeid, Amsterdam), Prof. Dr. G. Grote, Dr. T. Wäfler, and Mr. S. Little (ETH, Zurich), Dr. F. van Eijnatten, Dr. J. Goossenaerts, Dr. C. Pelletier, Drs. J. van de Bovenkamp, and Drs. Ir. R. Jongkind (TUE, Eindhoven), Dr. P. Orban, Mrs. M. Baldy, and Mrs. R. Steinmayr (RWTH, Aachen), Dr. J. Stahre and Mrs. E. Aresu (Chalmers University of Technology, Göteborg), Dr. J. Saari and Dr. T. Leskinen (Finnish Institute of Occupational Health, Helsinki), Prof. Dr. P. Groumpos and Dr. C. Stylios (University of Patras), Ir. R. van den Berg (Baan Development, Barneveld), Mrs. C. Reyneri, Mrs. L. Chiantore, Mrs. L. Medda, and Mrs. N. Epifani (Data Consult, Torino), Mrs. M. Sanseverino and Mr. A. Iuliano (CRFIAT, Torino), Mr. O. Tanninen (Finland Post, Helsinki), and Mr. J. Eskilsson and Dr. A. Johansson (Volvo, Göteborg).

The EU 5th Framework IST/PSIM project is a collaborative initiative with the IMS-based HUMACS project. The members of the HUMACS consortium are: Prof. E. Arai (Osaka University), Prof. H. Arisawa (Yokohama National University), Prof. K. Noro (Waseda University), Prof. A. Unal (Drexel University, Philadelphia), Prof. C. Slem (California Polytechnics State University), Prof. D. Leblanc (Ecole Polytechnique de Montreal), Mr. R. Mori and Mr. S. Yamada (Yamatake, Tokyo), Mr. N. Mitsui (Obayashi Corporation, Tokyo), Mr. Y. Tsuchitani (Kubota Corporation, Hyogo), Mr. K. Kawabata (Taizei Corporation, Tokyo), Mr. N. Kurosu (Toyota Motor Corporation), Dr. M. Boyer (CIRANO, Montreal), Mr. D. Bunker (Automation Specialties, Burlington), Dr. M. Frankel (SRI International, Menlo Park, California), Mr. R. Mahajan (Deneb Robotics, Michigan).

References

- [1] Drucker, P., The New Productivity Challenge, *Harvard Business Review*, **Nov-Dec** (1991) 69-79.
- [2] Quinn, J.B., *Intelligent Enterprise: A Knowledge and Service Based Paradigm for Industry*. The Free Press, New York, 1992.
- [3] Prahalad, C.K., and Hamel, G., Core Competence of the Corporation, *Harvard Business Review*, **January** (1990) 79-91.
- [4] Nonaka, I., and Takeuchi, H., *The Knowledge Creating Company*. Oxford University Press, New York, 1995.
- [5] Beardwell, I. and Holden, L., *Human Resource Management*. Pitman Publishing, London, 1994, p.694.
- [6] Vink, P., The Process of Improving Productivity by Worker Participation. In: *Proceedings of the 13th IEA Congress*, p.453-356, Volume I: Organizational Design and Management, Finnish Institute of Occupational Health, Helsinki, 1997.
- [7] Heller, F., Pusic, E., Strauss, G., and Wilpert, B., *Organizational Participation: Myth and Reality*. Oxford University Press, 1998.
- [8] European Foundation for the Improvement of Living and Working Conditions, *Communique July/August*, 1999, p.2.
- [9] Groumpos, P. & Krauth, J., Simulation in Industrial Manufacturing: Issues and Challenges. In: D. Fichtner, and R. Mackay (Eds.), *Facilitating deployment of information and communications technologies for competitive manufacturing*. The European Conference of Integration in Manufacturing—iIM, Dresden, 1997.
- [10] Riis, J.O. (Ed.), *Simulation Games and Learning in Production Management*. Chapman & Hall, London, 1995.
- [11] Forssen-Nyberg, M., Simulation Games – The State of the Art and the Future. In: *Proceedings of the 13th IEA Congress*, Volume I: Organizational Design and Management, Finnish Institute of Occupational Health, Helsinki, 1997, pp. 49-52.
- [12] Goossenaerts, J.B.M., A Framework for Connecting Work and Information Infrastructure. In: J.B.M. Goossenaerts, F. Kimura, and J.C. Wortmann, (Eds.), *Information Infrastructure Systems for Manufacturing*. Proceedings of the IFIP TC 5 Conference on the Design of Information Infrastructure Systems for Manufacturing (DIISM'96), Chapman & Hall, London, England, 1997.
- [13] Bruno, G., Reyneri, C., and Torchiano, M., Enterprise Integration - Operational Models of Business Processes and Workflow Systems. In: K. Kosanke, and J.G. Nell (Eds.), *Enterprise Engineering and Integration: Building International Consensus*. Research Report Esprit, Springer Verlag, Berlin, 1997.
- [14] Berg, R.J. van den, Eijnatten, F.M. van, Vink, P., & Goossenaerts, J.B.M., Leveraging Human Capital in Assembly Organizations: The Case for Participative Simulation. *Proceedings IST Conference*. Helsinki, 1999.
- [15] NRC (National Research Council), *Committee on Visionary Manufacturing Challenges, Visionary Manufacturing Challenges for 2020*, National Academic Press, Washington DC, 1998.
- [16] Hamel, G., and Prahalad, C.K., *Competing the Future*. Harvard Business School Press, Boston, 1994.
- [17] Coyne, K.P., Hall, S.J.D., and Clifford, P.G., Is Your Core Competence a Mirage?, *The Mc-Kinsey Quarterly* **1** (1997) 40-54.
- [18] Sprang, M. van, *Integral Assembly Improvement at Roberine*. Paper Presented at the Smart Assembling Workshop, Utrecht, 1999.

- [19] Eijnatten, F.M. van, and Zwaan, A.H. van der, The Dutch IOR Approach to Organizational Design: An Alternative to Business Process Re-engineering? *Human Relations* 51(3) (1998) 1-30.
- [20] Grote, G., A Participatory Approach to the Complementary Design of Highly Automated Work Systems. In: G. Bradley, and H.W. Hendricks (Eds.), *Human Factors in Organizational Design and Management - IV*. Elsevier, Amsterdam, 1994, pp. 115-120.
- [21] Grote, G, Wäfler, T., Ryser, C., and Windischer, A., KOMPASS: A Method to Aid Complementary Design of Automated Systems. In: *Proceedings of the First International Workshop on Intelligent Manufacturing Systems*, p.577-594, Lausanne, 1998.
- [22] Keizer, J.A., and Eijnatten, F.M. van (2000). A Holographic Process for Facilitating Workspace Solutions Within the Context of Holonic Organizational Decision Making. University of Technology, Faculty of Technology Management, Eindhoven, Deliverable 2 of Design of Workspace II, Brite-Euram Thematic Network BET2-516, 4th Framework Program Brussels.
- [23] Berg, R.J. van den, Grote, G., and Orban, P., Participative Simulation for Assembly Operations: A Problem Statement. ISATA, 99ADM059, 1999.
- [24] Eijnatten, F.M. van, Berg, R.J. van den, Vink, P., and Goossenaerts, J.B.M., Towards Integrated Intellectual Capital Management in Assembly Organizations: The Case for Participative Simulation. *Proceedings IMS Workshop Leuven*, 1999.
- [25] Vink, P., Eijnatten, F.M. van, Goossenaerts, J.B.M., Grote, G., Stahre, J., and Berg, R.J. van den, Making Technology Work in Intelligent Manufacturing by Participative Simulation.' *Proceedings IST Conference*, Helsinki, 1999.

Challenges in Dealing with Human Factors Issues in Manufacturing Activities

Shunji Yamada

Yamatake Corporation, Research and Development Headquarters
2-12-19 Shibuya, Shibuya-ku, Tokyo 150-8316, Japan
syamada@pres.yamatake.co.jp

Abstract. To whatever extent manufacturing facilities are automated, they still need humans with supreme authority over production management even in the new century. As sophistication in manufacturing systems increases, however, renewed operational and organizational issues have arisen associated with human resource management. This paper intends to give an overview of current topics and/or issues regarding the relationship between humans and machines in modernized manufacturing systems, and to provide a general idea on the challenges undertaken in an international joint research project developed under the IMS program for addressing human factors issues, based on a re-recognition of the significance of human roles in manufacturing activities.

1 Introduction

Is it realistic to seek ‘realization of an unmanned factory’ as a goal for advanced automatization of making things? Backed by a marked progress in electronics and computer technology since 1960s, is it appropriate to pursue to the utmost automatization or mechanization of all manufacturing activities including intellectual functions that humans have for decision-making? The answer to this question will be ‘no’.

In reality, consideration merely for stable operation of manufacturing at a work site equipped with an actual automated system would lead us to think of various situations that were not presumed at planning or design stage like: how to deal with occurrence of such abnormality as was absolutely beyond prediction; whether to guarantee safety operation even when some machinery gets out of work and malfunctions; or what to do with appropriate maintenance to cope with repair work or exchange of worn out parts. In addition, it would be fatal for competitiveness that no improvement to the manufacturing systems would be expected when humans are excluded from the manufacturing site.

Manufacturing at work site is not the only manufacturing activity. According to the International Research Council on Production and Machining, ‘manufacturing’ is defined as follows: “Manufacturing is a series of interrelated activities and operations involving the design, materials selection, planning, production, quality assurance, management and marketing of the products of the manufacturing industries”(Iwata et al[1982]). At present, computer systems are used in every stage of manufacturing activities in line with proliferation of information technology. Decades ago, the word ‘automation’ meant mechanization of manufacturing facilities in a narrow sense, but now it will include systematization of manufacturing activities as a whole following the above definition.

This paper intends to give an overview of current topics and/or issues regarding the relationship between humans and machines in modernized manufacturing systems. It provides a general idea on challenges undertaken in an international joint research project developed under the IMS program for addressing human factors issues, based on re-recognition of the significance of human roles in manufacturing activities.

2 What Is Human Factors?

When it comes to research on human characteristics differentiated from machines, so as to say, human aspects, or human elements, we face words 'human factors'. What does a phrase 'human factors' mean? Its interpretation used to be said to depend on situations. Also the definition has undergone gradual changes with time from meaning a narrow sense to covering broader meaning. The phrase can be used in daily conversation meaning all the factors related to humans like human aspects or human elements(Kuroda[1993]). It may be informative to first introduce as below its definition by Edwards(1985) cited in a book 'Human Factors in Flight' that is often referred to in study on human errors in aeronautics.

"Human factors is about people. It is about people in their working and living environments. It is about their relationship with machines and equipment, with procedures and with the environment about them. And it is also about their relationship with other people"(Hawkins[1992]).

It is seen that the applied technology of human factors is that it is concerned with optimizing the relationship between people and their activities by the systematic application of the human sciences, integrated within the framework of systems engineering, thereby ensuring effectiveness of the system, which includes safety and efficiency, and well-being of the individual. This is nothing but the theme the author is trying to tackle.

It is very much like a scientific word 'ergonomics' that is said to be composed of the Greek words *ergon* and *nomos* by the late Prof. Murrell in 1949. The professor then defined the word as "the study of man in his working environment"(Hawkins[1992]). Although human factors may give a rather broader meaning, what ergonomics means has been changed with time since then.

In 1949, the Ergonomics Society(<http://www.ergonomics.org.uk/>) was established in England as the first academic organization. In the U.S.A., Human Factors and Ergonomic Society(HFES)(<http://www.hfes.org/>) was founded in 1957. The International Ergonomics Association(IEA)(<http://www.iea.cc/>) was formed in 1959 as an international organization, and HFES got tied with this organization.

A recent noteworthy news in this connection is that the new definition of ergonomics was officially endorsed by IEA Board of Directors held in San Diego on July 29-30, 2000. The following is the new definition posted on the web site of IEA: "Ergonomics (or human factors) is the scientific discipline concerned with the understanding of the interactions among humans and other element of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance". Now ergonomics and human factors are completely equivalent to each other as scientific terms as well.

The new definition describes domains of specialization as "Practicing ergonomists must have a broad understanding of the full scope of the discipline, taking into account the physical, cognitive, social, organizational, environmental and other relevant factors". It is worthy to note that, besides physical ergonomics which has been the conventional mainstream such as human anatomical, anthropometric, physiological and biomechanical characteristics, specific description is given on cognitive ergonomics and organizational ergonomics as follows.

Cognitive ergonomics is concerned with mental processes, such as perception, memory, reasoning, and motor response, as they affect interactions of humans and other elements of a system. The relevant topics include mental workload, decision-making, skilled performance, human-computer interaction, human reliability, work stress and training as these may relate to human-system design.

Organizational ergonomics is concerned with the optimization of sociotechnical systems, including their organizational structures, policies, and processes. The relevant topics include communication, crew resource management, work design, design of working times, teamwork, participatory design, community ergonomics, cooperative work, new work paradigms, organizational culture, virtual organizations, telework, and quality management.

The scope defined can well be represented by the SHEL concept shown in Fig.1 which was proposed by F. H. Hawkins in the book cited earlier, where S, H, E, and L represent software, hardware, environment, and liveware, respectively.

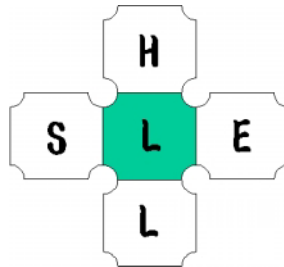


Fig. 1. Conceptual Model of Human Factors

Taking into consideration the discussion above, the academic field of human factors or ergonomics may well be said to encompass all phases of human, organizational, and social aspects in relation to humans. There may be no need any longer to care for whether a phrase human factors is used as a word for daily conversation or a scientific term. The subsequent discussion will deal with these terms without being aware of the situation in which they are used.

3 Topics on Human Factors in Manufacturing Activities

3.1 Human Error and Safety

Associated with human factors, issues of human error come up first, because 'to err is human.' A comprehensive and systematic study on human error may be best described in the book 'Human error: cause, prediction, and reduction'(Senders, Moray[1991]).

It is clear that human error is a potent and frequent source of hazard to human life and welfare, and to the ecosystems of Earth. Securing safety by eliminating human error is of crucial importance. The most typical research in this regard is made on operation and maintenance in aeronautics. A number of papers are published besides the book 'Human Factors in Flight' cited before (Takahashi[1993]; Maekawa, Senoo[1993]; Saito[1993]).

Likewise, there are several reports published on human factors issues in design and operation of nuclear plants (e.g. Kawano, Itoh, Kubota[1996]) which attracted attention worldwide triggered by grave accidents at a nuclear power plant in Three Mile Island in 1976 and at a Chernobyl nuclear power plant in 1986 (Kawano, Itoh, Kubota[1993]). In the case of chemical plants including petrochemical plants, the situation is the same as in nuclear plants in terms of safety control.

Regarding either nuclear plants or chemical plants, human machine interface for operation at a central control room equipped with advanced instrument and control systems is very much like that in a cockpit of an aircraft. Knowledge and experience in aeronautics that advances in terms of research on human factors have provided these process industries with useful suggestions or warnings. Recent topics concern computer control and human error in this industrial sector (Kletz, et al[1995]).

Human error occurring when humans are engaged in handling machines is so much complicated and delicate since it is closely related to a variety of factors humans have as living beings. Damasio's hypotheses (Damasio[1994]) disallow Descartes' dualisms that has guided traditional Western thought in terms of; e.g. mind vs body, or reason vs feeling. He insists "mental activity requires both brain and body proper." Rasmussen[1986] proposed a simplified model representing cognitive behavior of humans in such three layers as skill, rule, and knowledge bases. Preece et al[1994] emphasized the inherently multidisciplinary nature of human-computer interaction (HCI) pointing to a need to draw on concepts and skills from computer science, psychology, sociology, anthropology and industrial design. As he suggests, there is no panacea for HCI. We need to develop the right system to fit a specific purpose based on multidisciplinary studies.

Looking through research so far, however, would imply a trend focusing largely on technological design of L-H, L-S, and L-E interfaces on SHELL model referred earlier, specifically on design of human-machine interface (HMI) with partial consideration on physical environment (e.g., Nikkei Mechanical[1989]; Nishitani[1994]).

3.2 Ironical Aspects of Automation

The word 'automation' made its debut in 1960s with process automation that was advancing in nuclear, steel, and chemical plants. In the case of assembly industry, however, it was not before the advent of industrial robots in the 1970s that full-scale factory automation started. Until that time, the assembly sector used to be a labor-intensive industry where a number of workers were lined-up along conveyor belt. Automation technology made a great contribution to mass production by a decreasing number of workers. It truly freed direct workers from painful, dangerous, or drudgery jobs.

It appears in those earlier days that automation technology brought substantial benefits to humans. As sophistication in automated systems increases, however, the situation has changed quite a lot. The more intelligent and powerful an automated

system becomes, the more evident becomes the ‘lack of transparency of tools’(Kaiho, Tanabe[1996]) in the system. This means the plant operators’ increased reliance on the automated system for fulfilling their absolute duty of maintaining safe operation under any conditions. There arises the very basic problem inherent in machines: “The machine could not avoid possible malfunctions or failures.”

The most typical situation in this regard concerns relationship between plant operators and safety or emergency equipment, on which two different thoughts have remained for discussion: 1) Safety automation should be ultimately pursued due to less certain and less reliable nature of humans than machines; 2) Absolute safety could not be attained by machines however advanced they are, because of possible occurrence of unforeseeable incidents and unavoidable troubles with safety equipment itself(Terano[1983]).

Choice of the first strategy would mean that all authority rest with machines even under abnormal conditions while humans are being forced to behave as slaves, and the intelligent capability of humans associated with high level decision, experience, and inference is completely neglected. There would be no human dignity remaining with the plant operators.

Meanwhile, in safety equipment system designed on the second strategy, human decision would be in a higher position than machine’s, so the situation would be different from the first choice. The fact that humans are superior to machines, however, will directly mean heavier responsibility lying on humans. Although there may be a number of cases reported where high level decision by humans prevented malfunctioning by machines from causing serious incidents(e.g. Reason[1990]), no experts in plant operation would have such self-confidence as to make the right decision all the time. Humans in this case are exposed to everlasting stress for discovering possible troubles of safety equipment without knowing when these troubles are indeed occurring. This is another case of dehumanization albeit in a diametrically different sense from the first one. Furthermore, it means that automation which was once introduced for replacing unreliable humans now requires plant operators to complement the insufficiencies of systems design. Thus we truly have ‘Ironies of Automation’ (Bainbridge[1987]).

Under these circumstances, a lot of study has been made for alleviating mental burden or reducing human errors of process plant operators in the form of operation support systems and plant assets diagnosis systems(e.g. Fukuda[1997]; Matsuo, Minoshima, Takagaki[2000]). Continued intensive research is still called for seeking to curtail inhumane aspects of process automation.

3.3 Effective Allocation of Work between Humans and Machines

With the advent of microprocessors since the 1970s, mechatronics as it is typified by robotics, made a rapid progress. The most typical example may be seen at the workplace in assembly industry where such machines are installed. The relationship between humans and machines there is beyond the conventional level of HMI.

Sheridan[1992] mentioned the value of supervisory control relative to the degree of automation and task predictability referring to the all-or-none fallacy in terms of automation, and proposed a scale of ‘degrees of automation’ with a hint on where to stop automation in relation to human intervention. Inagaki et al proposed a design concept in which the degree of automation could be flexibly changeable depending on

changes in the situation such as the degree of emergency(Inagaki[1993]; Itoh, Inagaki, Moray[1999]).

According to Schweitzer[1996], the increasing ‘intelligence’ of machines makes such machines different from the previous ones and allows a more sophisticated cooperation with humans. Machines are being used as an extension of human skills and they are expected to communicate with their human users on an appropriate level. He points out the need to accept somehow a coexistence of technical systems with biological ones. This coexistence will certainly develop into cooperation.

He refers to issues regarding the allocation of functions and authority to man and machine, touching on a number of criteria for allocation proposed so far. The following criteria: left over, economic, comparison, and complementary, are briefly characterized as follows:

In the left over scenario, the degree of automation is as high as is technically feasible and only the tasks which cannot be automated are left over for humans. Following only economic criteria means that for each partial task the cheapest solution will be chosen. In the case of comparison, a task will be allocated to the one, machine or man, who performs it best.

The complementary approach assumes that man and machine have basically different capabilities which complement each other. The idea is to combine positive capabilities, the versatility of man and the consistency of the machine. Thus the objective is to assist an operator by providing him or her with an intelligent tool, and not to replace him or her by an automatic machine. Grote[1994] and her lab at ETH also are working on the complementary work allocation following this concept(Grote, Waefler, Weik[1996]).

The concept of the complementary approach is mentioned by Yamasaki[1996] as well with efforts on the pursuit of optimal allocation between humans and machines in automated systems. When it comes to a practical methodology, however, there may be no decisive solution available.

Given the policy clarified as the complementary scenario, increasing emphasis has been placed on effective collaboration between human and machine. In this respect, the concept of ‘Human-Centered Design’ proposed by Rouse is drawing attention. It is a process of assuring concerns, values, and perceptions of all stakeholders in the design efforts being considered and balanced(Rouse[1993]). Sawaragi is developing attractive research in this regard focusing on ecological aspects of human-machine interaction(Sawaragi[2001]).

3.4 Human-Centered Manufacturing Systems

The idea of human-centered systems or anthropocentric systems emerged in early 1990s as being more desirable as an extension from the coexisting and cooperative level of relationship between humans and machines.

The concept of ‘Anthropocentric Production Systems(APS)’ was defined in European Commission MONITOR-FAST program as “advanced manufacturing based on the optimal utilization of skilled human resources, collaborative industrial organization and adapted technologies.”(Wobbe[1992]) It points to the ‘insufficiency of competitive strategies with pure technology’, adding that “the dominant CIM concept of the 80’s was misleading, and often caused heavy misinvestments by one-sided technology oriented implementation measures”.

APS balances technology, organization and human factors in a holistic concept. It is a multi-dimensional concept of different elements. Action towards APS can be taken at various levels as: workplace; group work; factory organization; technical tools; organization and management; and education and training.

The European research project cited the MIT Study(Womack, Jones, Roos[1990]) as having proven by systematic comparison that Japanese car production had doubled productivity, not due to manufacturing technology, but to factory and work organization. The principle of production systems at Toyota Motor Corp. is well known as 'Toyota Production Systems(TPS)' in Japan, but is introduced overseas as 'lean production'(Womack, Jones, Daniel[1990]). Wobbe[1992] mentions that the lean production concept has a lot in common with APS, particularly in the organizational dimensions.

Actually, Toyota set up a corporate policy to realize a 'Worker Friendly Factory', made a thorough review on the work style in the final assembly shop, and succeeded in achieving a harmonious coexistence between workers and equipment, resulting in reduction of the line space and investment cost as well. The company also developed 'TVAL'(TOYOTA Verification of Assembly Line) for use in quantitative analysis of workload at each stage, which made a great contribution to upgrading of vehicle design, process and equipment planning(Kawamura, Niimi, Hisada, Kuzuha[1993]; Kako, Niimi, Eri[1995]; Niimi, Miyoshi, et al[1995]).

In the meantime, Nakazawa[1993] and his group have proposed since 1990 one of this type of manufacturing systems dubbed as 'Human-Oriented Manufacturing System(HOMS)', featuring the following three functional requirements: (1) a manufacturing system where workers can feel a sense of pleasure and pride through making things; (2) an economically feasible manufacturing system; (3) a human-friendly manufacturing system.

It is humans in HOMS who are leading players. It allows human intervention in a way that humans can easily operate a system even if it is automated. HOMS does not reject system automation. When humans are not at the work site, the system will operate automatically, thus making it possible to suppress capital investment, to improve productivity, and to produce highly value-added products. This will become a reality by taking advantage of human intervention with excellent human abilities such as tacit knowledge, creativity, and heuristics, leading to a manufacturing system with a higher level of intelligence, flexibility, and reliability of the system as a whole without additional investment.

The most important concept of HOMS is that manufacturing systems need to be designed such that direct workers can acquire a sense of pleasure and pride through making things. This type of systems will autonomously evolve through reflection of workers' ideas on improvement. Nakazawa summarized factors affecting workers' motivation for their work along with factors that would help produce pleasure of making things

3.5 Significance of Organizational Knowledge Creation

In a human-centered manufacturing factory or enterprise, an organizational strategy is of great importance for fully bringing out capabilities of individuals. This is effective for both a direct increase in organizational productivity and promotion of motivation

of individuals. There are quite a number of publications referring to this point(e.g. Maslow[1954]; Brown[1954]).

Prominent researchers in knowledge management describe knowledge as: "In an economy where the only certainty is uncertainty, the one sure source of lasting competitive advantage is knowledge"(Nonaka[1991]); "Knowledge should be seen as the resource for competitive advantage in current and future competition. They emphasize the management of knowledge at the organizational level, usually called 'core competence'."(Drucker[1991]) "A conventional interpretation of an organization as a mechanism for 'information processing' has a fundamental limitation; it does not really explain innovation." "The organization supports creative individuals or provides contexts for them to create knowledge. Organizational knowledge creation is to be understood as a process that 'organizationally' amplifies the knowledge created by individuals and crystallizes it as a part of the knowledge network of the organization."(Nonaka, Takeuchi[1995]) Furthermore, the Committee on Visionary Manufacturing Challenges at National Science Foundation(NSF), USA, also expresses in its report on 'Visionary Manufacturing Challenges for 2020' that "The basis of competition will be creativity and innovation in all aspects of the manufacturing enterprise."(Kramer[1999])

What matters is how to make into a reality such competitive edge. While these are all factors that are not as easily copied or reverse engineered, they evolve in the complex blend of corporate culture, strategies, and know-how(Berg, Eijnatten, Vink, Goossenaerts[1999]).

Fujimoto[1997] insists in this regard after years of research on Toyota Production System that the whole of activities in a manufacturing enterprise from development of new products through production and delivery could be described as a process of organizational knowledge creation and information transfer. Competitiveness of a manufacturing enterprise should be evaluated in such three levels as static competitive capability, capability for improvement, and capability for continually improving its capability, i.e. evolutionary capability. He tries to describe the Toyota system as an 'emergence process'(Kitamura, Kita[2001]) which grows without presuming proactive reasonableness. There is a report published with a somewhat sensational title of 'Decoding the DNA of the Toyota Production System'(Spear, Bowen[1999]), which says: "The Toyota story has been intensively researched and painstakingly documented, yet what really happens inside the company remains a mystery. Here's new insight into the unspoken rules that give Toyota its competitive edge". The authors summarize the tacit knowledge(Polanyi[1983]) that underlies the Toyota Production System in relation to its corporate culture.

The word 'participation' has been a key word for years in Japanese industries for promotion of organizational-knowledge-creating movements beginning in the context of QC circles. In the field of ergonomics, the concept of participatory ergonomics was proposed by Noro for assuring efficient utilization and integration of ideas and skills of experts and nonexperts working together(Noro, Imada[1991]). Now the participatory approach is widely recognized as an effective process for improvement of work methods(Vink, Peeters, Grundermann, et al[1995]).

There are quite a number of issues yet to be resolved to maximize the intellectual productivity in manufacturing industries while harmonizing it with the happiness and well-being of the individuals involved.

4 HUMACS Project and Its Significance

4.1 What Is HUMACS?

Against the backdrop mentioned in the preceding sections, there is an international research project under the IMS program named HUMACS that was originally proposed by Mori from Yamatake Corp.(Yamatake[1996]) IMS is an industry-led international R&D program established to develop the next generation of manufacturing and processing technologies(<http://www.ims.org/>). The acronym of the project concerned stands for “organizational aspects of HUMAN-MACHINE Coexisting Systems”, and directly tackles human and organizational issues encountered in manufacturing industries, which is associated with the fourth technical theme specified in the IMS Terms of Reference. Thus HUMACS aims to pursue a practical methodology to establish an optimum relationship between human factors and manufacturing facilities based on ergonomical, informational and sociotechnical studies on next generation manufacturing systems. Specifically, it is a challenge to solve the following problems that are commonly perceived by manufacturers in relation to the requirements mentioned earlier for desirable human-centered manufacturing systems:

- How to mobilize the human power most effectively for modernized manufacturing.
- How to preserve and enhance technical skills for manufacturing.
- How to exploit information technology to resolve sociotechnical problems in manufacturing enterprises.

After the preceding studies conducted in the Japan region, an international consortium was formed with its project proposal endorsed as an IMS international project in February, 1997. Under the umbrella name of HUMACS, its European version PSIM (Participative Simulation environment for Integral Manufacturing enterprise renewal) was defined. Its objectives are described in the project proposal(Vink[1999]) as follows:

“PSIM is a software environment for use in assembly operations and will be developed and pilot demonstrated in the project. PSIM uses a *Participative improvement process* involving specialized staff, management and production personnel. PSIM shows *Simulated assembly lines* in state-of-the-art ICT. PSIM is on *Integrated renewal*, which means that technological, organizational, and human factors are all concerned in optimization. It is focused on intelligent *Manufacturing* to assist human and technological creativity.”

Overall improvements in manufacturing performance will arise from efforts to ensure the optimum coexistence of human and machine based on the proper analysis and evaluation of the human factors. This research project was initiated to reflect such background and motives. It is not a mere pursuit of the optimum design of man-machine interfaces. It is a challenge to properly evaluate human factors from diverse angles, based on basic research in the organizational, sociological, and human engineering fields.

Yamatake Corp., Japan, serves as International Coordinator of the project accommodating four regions effectively involved as: Switzerland (1 partner), EU and Norway (11 partners), Japan (6 partners), and USA (1 partner). The names of the current participants are listed at the tail of this paper.

Increasingly active collaboration on development of technical themes is taking place in particular between Europe and Japan regions, leading to the first-phase completion of the project expected in March, 2002(Yamada, Vink, Grote, Brown[2001]).

4.2 Project Outline (Yamada, Vink[2000]; Yamada, Vink, Grote, Brown[2001])

The project of this kind which focuses on human-factors needs to encompass a wide range of study areas featuring interdisciplinary nature, but the research has been conducted on the three themes as:



Fig. 2. HUMACS Project Image

(1) optimum design of human-machine coexisting systems; (2) Sociotechnical factory organization; (3) Organized human-machine cooperative systems, where respective themes correspond to approaches from functional, organizational, and informational perspectives, while the human factors perspective is shared in common.

Fig.2 shows the project image drawn up from a slightly different viewpoint from the classification above with an aim to facilitate better understanding on the outputs from the whole project. In the center of the figure, ‘human-factors centered manufacturing enterprises’ is placed as a target object to be supported by a newly developed technical environment. The target enterprises are those where people involved give full play to their capability from every perspective through participation with full sense of fulfillment and satisfaction. A company uses humans that operate in networks and need motivation. For this support knowledge is needed (backbone) in the field of human science (ergonomics, work science, etc.), sociotechnique and advanced ICT. Also, platforms for knowledge sharing and application are needed in design and modeling, concurrent & participative simulation and human modeling and ergonomics.

Their activities inherently lead to continually lasting improvement and innovation of work methods. The arrows running diagonally upward from left to right indicate a spiral-up image of this movement. The platforms in the bottom are considered tools to uphold these most human-like activities. A briefing is given below on practical methodologies or tools in this regard developed so far within the HUMACS project.

(1) Human-modeling and ergonomics platform

There are two practical methodologies being developed for supporting the bottom layered platform. One is 'Info-Ergonomics' modeling(Arisawa, Imai[1996]; Arisawa[1997]), and the other psychophysiological evaluation.

Info-Ergonomics technology, which means a fusion of IT and ergonomics, has been created at Arisawa Lab at Yokohama National University on a firstly developed multi-media database environment dubbed Real World Database(RWDB) with cooperation from Kubota Corp.(Arisawa, Tomii, Yui, Ishikawa[1995]; Arisawa, Tomii, Salev[1996]) Info-Ergonomics provides a markerless image retrieval from data in the DB and CG replay of dynamic behavior based on interpretation of meanings, shapes, and actions of objects concerned(Tomii, Salev, Imai, Arisawa[1998]; Salev, Tomii, Arisawa[1998]; Tomii, Kobayashi, Arisawa[1999]). Human mock-ups are becoming precise enough to represent physical characteristics of individuals, thereby being applicable even to the scene of medical treatment(Imai, Tomii, Arisawa[2000]; Sato, Nagano, Tomii, Arisawa, Sakai[2001]). Its industrial application is highly expected to the analysis and evaluation from diverse perspectives of kinesiology and physiologic rationality in working conditions at the workplace, contributing to pursuit of the optimal condition for human-machine coexistence(Arisawa[1997]; Imai, Kobayashi, Masutani, Arisawa[1997]; Imai, Salev, Tomii, Arisawa[1998]; Arisawa, Imai[2001]).

The psychophysiological evaluation is being conducted at Noro Lab at Waseda University in such a way as, using a newly-developed virtual working environment(Noro, Oyama, Tanaka, Fujinami[2000]), desk work in office is analyzed and evaluated from ergonomic viewpoints through psychophysiological experiments with relevant physiological data taken by means of several measuring systems(Takao, Noro[1999]; Noro, Tanaka[2001]; Noro, Teraoka, Hashimoto[2001]).

(2) Concurrent and participative simulation platform

One of the system environments serving as a concurrent and participative platform is a simulation environment developed by Arai Lab at Osaka University. It is a challenge to make one step forward to implement the specifications defined in HOMS mentioned earlier. A prototype simulation environment with a real-time scheduler called Production Auction System(PAS) was developed with an active database as its nucleus surrounded by distributed autonomous cells(Arai, Wakamatsu, Takagi, Uchiyama, Takata[1996]; Arai, Shirase, Wakamatsu, Murakami, Takata, Uchiyama[1997]). Attention has been paid to build a mechanism inside the cells which possesses humanity equipped with emotions which machines do not have, and to amalgamate higher flexibility like the ability for adaptive collaboration between cells(Arai, Wakamatsu, Takata, Shirase[1999]; Shirase, Nagafune, Wakamatsu [2000]). Furthermore, a trial is being made to apply this prototype to simulation of

manufacturing operation of printed wiring board (PWB) assembly lines at Yamatake Corp. in an attempt to improve overall productivity by overcoming the lack of flexibility inherent to conventional automatic assembly machines that are designed suitable solely for mass production(Chikano[2001]).

A participative simulation environment being developed in PSIM basically falls in this category of the platform. The goal of participative simulation is to enable workers to exert direct influence over the product and process designs by bringing in their tacit knowledge, to combine it with expert knowledge, and to put the blend of both insights to the test, and eventually into practice(Berg, Grote, Orban[1999]; Eijnatten, Berg, Goossenaerts[1999]; Berg, Eijnatten, Vink, Goossenaerts[1999]; Vink, Eijnatten, Goossenaerts, Grote, Stahre, Berg[2000]).

The deliverables from PSIM mostly still remain inside the project, in which the expert knowledge is provided by incorporated tools reflecting expertise in ergonomics and sociotechnical systems design(STSD). A paper and pencil version of the environment has been tested and evaluated at European test sites.

(3) Design and modeling platform

Itoh Lab at Sophia University and Yamatake Corp. have jointly developed an Integrated Collaboration and Concurrent Engineering Environment(ICCEE) (Kumagai, Kawabata, Hasegawa, Itoh[2000]; Itoh, Kumagai et al[2001]). A collaboration task is a process which involves diversified individuals and/or organizations. Each participant in the process has different roles and perspectives in the process, while sharing a common goal. The concept of Multi-Context Map(MCM) was created for facilitating building workflow for this type of collaborative work, and plays a pivotal role in ICCEE(Kumagai, Hasegawa, Kawabata, Itoh[2000]; Hasegawa, Itoh, Kumagai[2001]). ICCEE is intended to improve the efficiency of the whole systems design process in a concurrent manner. Collaborative work to be supported is described in MCM, and navigators/converters are developed for conversion between MCM and IDEF0 as well as for facilitating smooth transfer from MCM to performance evaluation by GPSS simulator(Kumagai, Itoh[1998]; Hasegawa, Itoh, Kumagai[2000]).

4.3 Significance of HUMACS

The significance of the HUMACS project firstly is its attitude toward frontally tackling human-factors issues in the manufacturing industry. Compared with HOMS mentioned earlier which ends up with requirement definition for human-factors centered manufacturing systems, HUMACS, as was briefed in the preceding section, has proposed a specific system architecture. Furthermore, it has made one step forward to implementation through development and industrial application of a prototype of the Production Auction System. In the area of physical ergonomics, it is of great technical significance for Info-Ergonomics to have realized markerless identification of semantics and reflection of individual features on human mock-up. Industrial application of this technology along with psychophysiological evaluation will be sure to facilitate an increase in comfort and safety of workers as well as higher efficiency of work through optimized design of workers' movements, creation of guidelines based on experts' performance analysis, etc.

Besides, in the course of research on human science as a backbone technology in HUMACS, it is worth noting that a concept of ‘pleasure of making things’ in HOMS has been deepened through research on psychological interaction between human and machine at the manufacturing site. The task developed a ‘human-machine coexisting model’ in the early stage of the project which systematically visualizes psychological process of a worker who faces machines, clarifying positions of unconsciousness, intuition, congeniality with machine, and so forth(Mori[1994]). The ‘Mori model’ has been making a lot of contribution to global sharing of a structural idea of human-machine coexisting systems. And further research led to the creation of a model on ‘operator’s adaptation process’, thereby deepening a sense of happiness and a value of living(Mori[1995]; Mori[1997]). These achievements are included in the task of ‘Image Playback Model’ as shown in Fig.3.

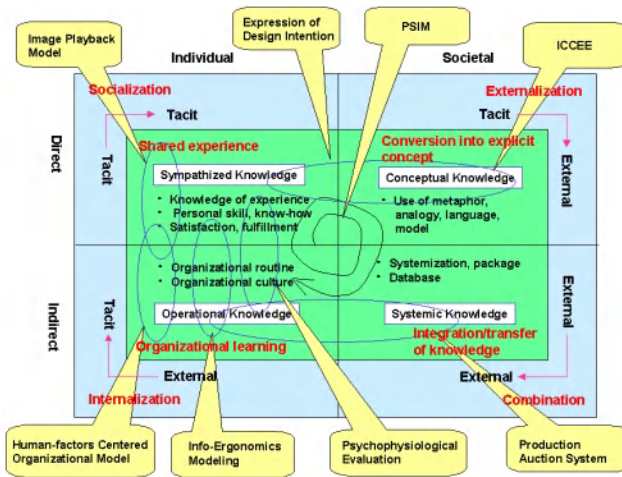


Fig. 3. HUMACS Tasks Mapped on SECI Model

Secondly, HUMACS has comprehensively tackled on how to support organizational knowledge creation. PSIM intends to provide a specific tool for positively “supporting approaches to amplify knowledge in the upward spiral knowledge-creating process: from the individual, over the team to the organization”.

The key factors for a breakthrough in this approach are: participation, simulation, ICT architecture, and ontology. Specifically, a newly developed PSIM ontology facilitates interoperation and communication between applications (e.g., ERP, MES, Sociotechnical tools, Ergonomy tools(Bovenkamp, et al[2001])) and it allows us to integrate the related knowledge acquisition, discovery, and modeling efforts of the end-users within the context of their enterprise(Goossenaerts, Pelletier[2001]).

ICCEE presented earlier is aimed at modeling of collaboration between stakeholders including experts and non-experts, seeking for higher efficiency in collaborative work, and thus will contribute to externalization of tacit knowledge. The

same is true for another task of addressing 'Expression of Design Intention' which is a challenge to externalize such designers' intention as is inexpressible by conventional means like documents and drawings(Arai, Okada, Iwata[1991]; Arai, Shirase, Wakamatsu[1998]; Arai, Akasaka, Shirase, Wakamatsu[2000]). A task led by Toyota Motor Corp. of 'Human-factor Centered Organizational Model' focuses on a concept of a 'living organization' that is closely related to the creation of workplace climate and corporate culture(Kurosu, Yamada[2001]).

Looking over the research tasks undertaken within the project will lead us to plot them on Nonaka's SECI model (Nonaka, Takeuchi[1995]) as shown in Figure 3. As is well known, the left and right halves represent the tacit and explicit knowledge dimensions, respectively. There are a growing number of research themes reported on handling information and knowledge on the explicit knowledge dimension in line with progress of ICT with respect to achieving more efficient and effective knowledge sharing, transfer, and transformation. When it comes to research on the handling of knowledge or information on the tacit dimension, however, it is far behind that of the externalized dimension; especially, this is true for the practical methodology to facilitate knowledge-creation process through the left half of the model. The figure may well be said to demonstrate the significance of the HUMACS project making marked contribution to a continuous and dynamic interaction between tacit and explicit knowledge dimensions with the PSIM environment serving as an engine of the entire spiraling process.

The final point to be marked as a characteristic of HUMACS is its nature of an international joint research project. It is only reasonable that appropriate handling of human factors will be affected to a great extent by nationality, tradition, and culture to which the people concerned are related.

The organizational knowledge creation is closely related to workplace climate and corporate culture, or, in other words, traditional ways of organizational operation which show sharp contrast between Western and Japanese styles. In short, while Western engineers tend to emphasize explicit knowledge based on a view of an organization as a mechanism for information processing, the Japanese tend to stress tacit knowledge, where team working has prevailed so as to encourage socialization of tacit knowledge as well as organizational learning in internalization in the course of knowledge conversion process. In addition, participation by direct workers in improvement to work methods is more commonly adopted in Japanese organizations. Likewise, mutual interactions between different regions are expected in the area of physical and cognitive ergonomics as well(Yamada, Vink, Grote, Brown[2001]).

5 Expectation to Future Development

Current topics and issues over the relationship between humans and machines in modernized manufacturing systems have been overviewed, and a general idea on challenges being undertaken in an international joint research project has been provided. Extensions to the future development will be viewed as follows:

First, an extension of the HUMACS project is planned on a European initiative with a tentative name of HUMACS2 focusing on the development of a digital design factory as a tool to improve human performance in assembling(Vink[2001]). The technologies developed in HUMACS like digital human modeling, virtual reality

training, and mixing real humans into virtual reality will be further enhanced in the new project toward actual industrial application, leading to projected reduction of inefficient human handlings, musculoskeletal injuries, training time on assembling, and to more motivated workforce by testing team functioning in the digital factory.

Secondly, a need for corporations to reach new levels of flexibility and adaptability is proposed in the 'Vision of Manufacturing in 2020' compiled by NSF referred to earlier(Kramer[1999]): "The basis for competition will be creativity and innovation in all aspects of the manufacturing enterprises." The need emphasizes the significance of social and organizational structures to be knowledge-based, dynamic, fluid, and globally distributed. This signals the growing significance of properly addressing human factors issues in manufacturing in the new century.

More intensive research is further expected both on the extension of the HUMACS project and on how to deal with a broader range of human factors, based on the re-recognition of the significance of human roles in manufacturing activities.

Acknowledgements. The author as coordinator of the HUMACS project has described an overview of the international HUMACS project on behalf of the consortia involved.

Its current members are: Prof. G. Grote, Mr. T. Waefer, and Mr. S. Little (ETH, Switzerland), Mr. H. Sasajima, Mr. Y. Shiote and Ms. M. Chikano (Yamatake Corp., Japan), Mr. R. Mori and Mr. S. Yamada (Yamatake Industrial Systems, Japan), Mr. N. Kurosu (Toyota Motor Corp., Japan), Prof. H. Arisawa (Yokohama National Univ., Japan), Prof. E. Arai (Osaka Univ., Japan), Prof. K. Noro (Waseda Univ., Japan), Prof. K. Itoh (Sophia Univ., Japan), Dr. P. Vink and Ir. G. van Rhijn (TNO Arbeid, EU), Dr. F. van Eijnatten and Dr. J. Goossenaerts (TUE, EU), Dr. P. Orban and Ms. M. Baldy (RWTH, EU), Dr. J. Stahre (Chalmers Univ. of Tech, EU), Dr. J. Saari and Dr. T. Leskinen (Finish Institute of Occupational Health, EU), Prof. P. Groumpos (Univ. of Patras, EU), Ir. R. van den Berg and Dr. A. Zwegers (Baan, EU), Ms. C. Reyneri (Data Consult, EU), Ms. M. Sanseverino and Mr. A. Iuliano (CRFIAT, EU), Mr. O. Tan-ninen (Finland Post, EU), Mr. J. Eskilsson (Volvo, EU), Mr. R. Brown (Delmia, USA)

In addition, the former members with contribution in the early stage are: Dr. M. Boyer (CIRANO, Canada), Prof. D. Leblanc (Ecole Polytechnique de Montreal, Canada), Mr. D. Bunker (Automation Specialties, Canada), Mr. N. Mitsui (Obayashi Corp., Japan), Mr. Y. Tsuchitani (Kubota Corp., Japan), Mr. K. Kawabata (Taisei Corp., Japan), Prof. A. Unal (Drexel Univ., USA), Prof. C. Slem (California Polytechnics State Univ., USA), Dr. M. Frankel (SRI International, USA).

References

- Arai, E., Akasaka, H., Shirase, K., Wakamatsu, H., 2000, Description Model of Designers' Intention in CAD System and Application for Redesign Process, JSME International Journal Series C, Vol.43, No.1, 2000, pp.177-182,
- Arai, E., Okada, K., Iwata, K., 1991, Intention Modeling System of Product Designers in Conceptual Design Phase, Manufacturing Systems, Vol.20 (1991), No.4, , pp.325-333,

- Arai, E., Shirase, K., Wakamatsu, H., Murakami, Y., Takata, M., Uchiyama, N., 1997, Role of Production Simulation for Human Oriented Production Systems, Proceedings of the 14th International Conference on Production Research, Osaka, August 4-8, 1997, pp.758-761,
- Arai, E., Wakamatsu, H., Takagi, S., Uchiyama, N., Takata, M., 1996, Auction Based Production System through Active Database System, Proceedings of PCM'96, Seoul, October 29-31, 1996, pp.371-376,
- Arai, E., Wakamatsu, H., Takata, M., Shirase, K., 1999, Human Oriented Production System Architecture, The 15th International Conference on Production Research, Univ. of Limerick, 9-12 August 1999, pp.395-398,
- Arai, E., Shirase, K., Wakamatsu, H., 1998, CAD with use of Designers' Intention, Proceedings of The Third IFIP Working Group 5.2 Workshop on Knowledge Intensive CAD, Tokyo, December 1-4, 1998, pp.21-30,
- Arisawa, H., 1997, Application of Info-Ergonomics to Design of Human-Machine Coexisting Systems, Proc. of Technical Conference of IMS Japan, September 10-11, 1997, MSTC/IMS Promotion Center, pp.63-66, (in Japanese)
- Arisawa, H., Imai, S., 1996, Working Simulation based on Info-Ergonomics and Multimedia Database Concept Design, 1996 Japan-U.S.A. Symposium on Flexible Automation, July 8-10, 1996, Boston,
- Arisawa, H., Imai, S., 2001, Mediator-Based Modeling of Factory Workers and Their Motions in the Framework of Info-Ergonomics, Human Friendly Mechatronics (Selected Papers of ICMA2000), May 2001, Elsevier, pp.395-400,
- Arisawa, H., Tomii, T., Salev, K., 1996, Design of Multimedia Database and a Query Language for Video Image Data, IEEE Multimedia Systems'96, Hiroshima, June 17-21, 1996,
- Arisawa, H., Tomii, T., Yui, H., Ishikawa, H., 1995, Data Model and Architecture of Multimedia Database for Engineering Applications, IEICE Transactions of Information and Systems, Vol.E78-D, No.11, November 1995, pp.1362-1368
- Bainbridge, L., 1987, Ironies of Automation, In J. Rasmussen et al., New Technology and Human Error, Wiley, pp.271-283,
- Berg, R.J. van den, Eijnatten, F.M. van, Vink, P., Goossenaerts, J.B.M., 1999, Leveraging human capital in assembly organisations: The case for participative simulation, Proceedings IST Conference Helsinki, Brussels 5th Framework Program,
- Berg, R.J. van den, Grote, G., Orban, P., 1999, Participative simulation for assembly operations: A problem statement, ISATA: 99ADM059,
- Bovenkamp, M. van den, Jongkind, R., Rhijn, G. van, Eijnatten, F. van, Grote, G., Lehtela, J., Leskinen, T., Little, S., Vink, P., Wafler, T., The E/S tool: IT-support for ergonomic and sociotechnical system design, Proceedings of ER2001, Yokohama, November 2001, to appear,
- Brown, J. A., 1954, The Social Psychology of Industry, , Penguin Books,
- Chikano, M., 2001, A Study on Real-Time Work Support on a Printed Circuit Board Assembly Line, Proc. of Technical Conference of IMS Japan, July 10-11, 2001, MSTC/IMS Promotion Center, pp.61-64, (in Japanese)
- Damasio, A. R., 1994, Descartes' Error, , A Grosset/Putnam book,
- Drucker, P. F., 1991, The New Productivity Challenge, Harvard Business Review November-December 1991, pp.69-79,
- Eijnatten, F.M.van, Berg, R.J. van den, Goossenaerts, J.B.M., 1999, Towards participatory intellectual capital management in assembly organisations: The case for participative simulation, Proceedings of the Second International Workshop on Intelligent Manufacturing Systems, Leuven, pp.857-863,
- Eijnatten, F.M.van, Berg, R.J. van den, Goossenaerts, J.B.M., 2000, ICT-supported participative simulation to enable integrated intellectual capital management, E-business: Key issues, applications and technologies, Amsterdam: IOS Press, pp.659-665,

- Fujimoto, T., 1997, Evolution of Manufacturing Systems, Yuhikaku, , (in Japanese)
- Fukuda, Y., 1997, IMAC: A Tool for Supporting Nonstationary Plant Operation, Instrumentation, Vol.40, No.5, 1997, pp.44-49, (in Japanese)
- Goossenaerts, J., Pelletier, C., 2001, Enterprise Ontologies and Knowledge Management, Proceedings of ER2001, Yokohama, November 2001, to appear,
- Grote, G., 1994, A participatory approach to the complementary design of highly automated work systems, G. Bradley and H. W. Hendrick, eds., Human factors in organizational design and management, Amsterdam, Elsevier,
- Grote, G., Waefler, T., Weik, S., 1996, Complementary Function Allocation as Basis for Safety in Process Control, 5th European conference on cognitive science approaches to process control, pp.8-17,
- Hasegawa, A., Itoh, K., Kumagai, S., 2000, Collaboration Task Analysis by Identifying Multi-Context and Collaborative Linkage, Concurrent Engineering Research and Applications, Vol.8, No.1, March 2000, pp.61-71,
- Hasegawa, A., Itoh, K., Kumagai, S., 2001, Collaboration Engineering Analysis Method for Manufacturing System, Advances in Concurrent Engineering, June 2001 (to appear in CE 2001), to appear,
- Hawkins, F. H., 1992, Human Factors in Flight, , Ashgate,
<http://www.ergonomics.org.uk/>, , The Ergonomics Society,
<http://www.hfes.org/>, , Human Factors and Ergonomic Society,
<http://www.iea.cc/>, , International Ergonomics Association,
<http://www.ims.org/>, , Intelligent Manufacturing Systems,
- Imai, S., Kobayashi, M., Masutani, K., Arisawa, H., 1997, Design of Time-Space Objects in Design of a Database of Factory Work, Database System 1997. 7. 16, pp.365-370,
- Imai, S., Salev, K., Tomii, T., Arisawa, H., 1998, Modelling of Working Processes and Working Simulation based on Info-Ergonomics and Real World Database Concept, Proc.of 1998 Japan-U.S.A. Symposium on Flexible Automation, July 1998, pp.147-154,
- Imai, S., Tomii, T., Arisawa, H., 2000, Human Body/Motion Modeling and Database Design based on the Mediator Concept, Transaction of IPSJ: Database, Vol.41, No.SIG 1 (TOD 5), January 2000, pp.100-108,
- Inagaki, T., 1993, For whom is automation?, J. SICE Vol.32, No.3, March 1993, pp.181-186, (in Japanese),
- Itoh, M., Inagaki, T., Moray, N., 1999, Trust in Situation-Adaptive Automation for Systems Safety, Tech. J. SICE, Vol.35, No.7, pp.943-950 (1999), (in Japanese),
- Itoh, K., Kumagai, S., et al, 2001, An Integrated Environment for Development of Distributed Production Support Systems, Proc. of Technical Conference of IMS Japan, July 10-11, 2001, pp.49-52, (in Japanese)
- Iwata, K., et al, 1982, Manufacturing Systems Science, Corona Sha, , (in Japanese)
- Kaiho, H., Tanabe, F., 1996, Human Error, Shinyosha, (in Japanese)
- Kako, H., Niimi, A., Eri, Y., 1995, On the Development of TVAL(Toyota Verification of Assembly Line) and Its Application, Toyota Technical Review Vol.44, No.2, Mar. 1995, pp.77-82, (in Japanese)
- Kawamura, T., Niimi, A., Hisada, N., Kuzuhara, T., 1993, Coming Worker Friendly Factory, Toyota Technical Review Vol.43, No.2, Nov. 1993, pp.86-91, (in Japanese)
- Kawano, R., Itoh, J., Kubota, R., 1993, Human Factors in Operation of Nuclear Plants, Journal of IEE Japan Vol.113, No.10, October 1993, pp.821-827, (in Japanese)
- Kawano, R., Itoh, J., Kubota, R., 1996, Safe Operation of A Nuclear Plant and Human Factors, J. SICE Vol.35, No.8, August 1996, pp.601-605, (in Japanese)
- Kitamura, S., Kita, H., 2001, Emergence System, J. SICE Vol.40, No.1, January 2001, pp.94-99, (in Japanese)
- Kletz, T., et al, 1995, Computer Control and Human Error, , Institute of Chemical Engineers,

- Kramer, B., 1999, Visionary Manufacturing Challenges for 2020, Presentation to IMS Forum'99, December 1, 1999,
- Kumagai, S., Itoh, K., 1998, Designing Collaborative Work in IDEF0 Using Interface Model, *Journal of Concurrent Engineering*, Vol.6, No.9, December 1998, pp.333-343,
- Kumagai, S., Kawabata, R., Hasegawa, A., Itoh, K., 2000, Integrated Environment for Collaboration Engineering by Collaboration Interface Model, *Proceedings of IDPT 2000*, June 2000,
- Kumagai, S., Hasegawa, A., Kawabata, R., Itoh, K., 2000, Building Workflow for Collaboration Task Using Multi-Context Map , *Transactions of the SDPS (Society for Design and Process Science)*, Vol.4, No.3, Sept 2000, pp.49-62,
- Kuroda, I., 1993, History of Research on Human-factors and Its Recent Trend, *Journal of IEE Japan* Vol.113, No.10, October 1993, pp.806-808, (in Japanese)
- Kurosu, N., Yamada, S., 2001, Living Manufacturing Systems with Living Organizations, *Proceedings of ER2001*, Yokohama, November 2001, to appear,
- Maekawa, H., Senoh, M., 1993, Human Factors in Aeronautics, *Journal of IEE Japan* Vol.113, No.10, October 1993, pp.816-818, (in Japanese)
- Maslow, A. H., 1954, *Motivation and Personality*, Harper & Row, Publishers,
- Matsuo, T., Minoshima, H., Takagaki, J., 2000, Quality Stabilization by Improving the Controllability of Polyolefin Production System, *Proc. of Technical Conference of IMS Japan*, July 11-12, 2000, CD-rom, (in Japanese)
- Mori, R., 1994, Human Factors in CIM Environment, *Proc. of the Asian Control Conference*, Tokyo, July 27-30, 1994, pp.271-273,
- Mori, R., 1995, A Manufacturing System Featuring on Workers' Subjective Well-being, *Proceedings of FA Symposium SICE*, June 20, 1995, pp.13-17, (in Japanese)
- Mori, R., 1997, Humans' Subjective Well-being in Manufacturing Industries, *J. SICE* Vol.36, No.4, April 1997, pp.295-298, (in Japanese)
- Nakazawa, H., 1993, Human-Oriented Manufacturing Systems, The 200th Lecture Note, The Japan Society for Precision Engineering, 1993-10-27, (in Japanese)
- Niimi, A., Miyoshi, K., Ishii, T., Araki, T., Uchida, K., Ota, I., 1995, Establishment of Autonomous Complete Process Planning for Autonomous Assembly Line, *Toyota Technical Review* Vol.44, No.2, Mar. 1995, pp.83-88, (in Japanese)
- Nikkei Mechanical, 1989, Human Interface, *Nikkei Mechanical* 1989.1.23, pp.108-115, (in Japanese)
- Nishitani, H., 1994, Human-Computer Interaction in the New Process Technology, *Proceedings of PSE94*, Kyongju, Korea, May 30-June 3, 1994, pp.1367-1375
- Nonaka, I., 1991, The Knowledge-Creating Company, *Harvard Business Review* November-December 1991, pp.96-104,
- Nonaka, I., Takeuchi, H., 1995, *The Knowledge-Creating Company*, Oxford Press,
- Noro, K., Imada, A., 1991, *Participatory Ergonomics*, Taylor & Francis,
- Noro, K., Tanaka, R., 2001, Participatory approach for development of Japanese new VDT guidelines, *Workshop of the Dutch Ergonomics Association on participatory ergonomics*, February 2001,
- Noro, K., Oyama, H., Tanaka, R., Fujimaki, G. , 2000, Creating the seating clinic and design studio : a participatory approach of seating ergonomics, *Human Factors and Ergonomics, International Ergonomics Association*, August 2000, CD-rom,
- Noro, K., Teraoka, T., Hashimoto, A., 2001, History of Working Posture in Japan, 2nd *International Conference on the History of Occupational and environmental Prevention*, September 2001, Norrkoping Conference 2001,
- Polanyi, M., , 1983, *The tacit dimension*, Gloucester, Mass.,
- Preece, J., et al, 1994, *Human-Computer Interaction*, , Addison-Wesley Publishing Co.,
- Rasmussen, J., 1986, *Information processing and human-machine interaction*, North-Holland,
- Reason, J., 1990, *Human Error*, Cambridge University Press,

- Rouse, W. B., 1993, Human-Centered Design: Concept and Methodology, *J. SICE* Vol.22, No.3, March 1983, pp.187-192,
- Saito, M., 1993, Dealing with Human Factors in Maintenance Work for Airplanes, *Journal of IEE Japan* Vol.113, No.10, October 1993, pp.819-820, (in Japanese)
- Salev, K., Tomii, T., Arisawa, H., 1998, Extracting Event Semantics from Video Data Based on Real World Database, *Proceedings of ER98 Workshops on Data Warehousing and Data Mining, Mobile Data Access, and Collaborative Work Support and Spacio-Temporal Data Management*, Singapore, November 1998, Springer-Verlag, pp.553-567,
- Sato, T., Nagano, S., Tomii, T., Arisawa, H., Sakai, N., 2001, Design and Construction of Human-Body Motion Database Using Bone-Based Human Model, *Transaction of IPSJ: Database*, Vol.42, No.SIG 1 (TOD 8), January 2001, pp.92-102, (in Japanese),
- Sawaragi, T., 2001, An Ecological Approach to Human-Centered Design, *Lecture Note, Human Interface Symposium 2001*, Osaka, October, 2001, (in Japanese),
- Schweitzer, G., 1996, Challenges in Mechatronics and Robotics, *Keynote at Intl Conf. on Motion and Vibration Control*, Chiba, Sept. 1-6, 1996,
- Senders, J.W., Moray, N.P., 1991, *Human error (cause, prediction, and reduction): analysis and synthesis*, Lawrence Erlbaum Associates, Publishers,
- Sheridan, T.B., 1992, *Telerobotics, Automation, and Human Supervisory Control*, MIT Press,
- Shirase, K., Nagafune, N., Wakamatsu, H., Arai, E., 2000, Human Oriented Production Management Considering Working Satisfaction, *Proceedings of PCM 2000*, Detroit, September 6-8, 2000, pp.733-738,
- Spear, S., Bowen, H. K., 1999, Decoding the DNA of the Toyota Production System, *Harvard Business Review* September-October 1999, pp.97-106,
- Takao, H., Noro, K., 1999, A Study on Learning of Direction of Sound Images in Virtual Reality, *Proc. of Joint System Conference* April 1999,
- Terano, T., 1983, Restoration of Human Power and Fuzzy Engineering, *J. SICE* Vol.22, No.1, January 1983, pp.2-5, (in Japanese)
- Tomii, T., Salev, K., Imai, S., Arisawa, H., 1998, Human Modelling and Design of Spatio-Temporal Queries an 3D Video Database, *Proc. of IFIP 2.6 Working Conference VISUAL DATABASE SYSTEMS-4*, L'Aquila, Italy, May 27-29, 1998, Chapman & Hall, pp.317-336,
- Tomii, T., Kobayashi, M., Arisawa, H., 1999, Design of Real World Scene Database with Mapping to Virtual CG Space, *IEICE Transactions*, D-I Vol.J82-D-I, No.1, January 1999, pp.211-222,
- Vink, P., 1999, PSIM Annex 1-"Description of Work", *IST Program Proposal No. IMS-1999-00004*, October 1999,
- Vink, P., 2001, The digital design factory as a tool to improve human performance in assembling, *HUMAN Coexisting Systems 2 (HUMACS2)*, Abstract proposal to IMS, June 2001,
- Vink, P., Eijnatten, F.M. van, Goossenaerts, J.B.M., Grote, G., Stahre, J., Berg, R.J. van den, 2000, Making technology work in intelligent manufacturing by participative simulation , *Proceedings of the XIVth IEA and 4th HFES meeting* , pp.2571-2574
- Vink, P., Peeters, M., Grundermann, R.W.M., Smulders, P.G.W., Kompier, M.A.J., Dull, J., 1995, A participatory ergonomics approach to reduce mental and physical workload, *International Journal of Industrial Ergonomics* 15 (1995), pp.389-396,
- Wobbe, W., 1992, What are anthropocentric production systems? What are they a strategic issue for Europe?, *Science and Technology Policy, Final Report*, EUR 13968, 1992, Commission of the EC,
- Womack, J., Jones, R., Roos, D., 1990, *The Machine That Changed the World* , Harper Perennial,
- Yamada, S., Vink, P., 2000, Organizational Aspects of Human-Machine Coexisting Systems, *Proceedings of PCM 2000*, Detroit, September 6-8, 2000, pp.981-988,

- Yamada, S., Vink, P., Grote, G., Brown, R., 2001, Promotion of HUMACS as an International Project, Proceedings of IMS Project Forum 2001, Ascona, October 8-10, 2001, to appear,
- Yamasaki, H., 1996, Optimized Allocation of Work between Human and Machine in Automated Systems, J. SICE Vol.35, No.8, August 1996, pp.571-574, (in Japanese)
- Yamatake Corp., 1996, Organizational Aspects of Human-Machine Coexisting System, HUMACS Project Proposal Version 2.2, Intelligent Manufacturing System Program, October 1996,
- Yukimachi, T., 1996, Operation Support Technology for Preventing Human Errors, J. SICE Vol.35, No.8, August 1996, pp.586-591, (in Japanese)

Extracting E-R Models from Collaboration Analysis Methods, MCM, and CLM

Manabu Kamimura¹, Naoyuki Kato¹, Ryo Kawabata¹,
Satoshi Kumagai², and Kiyoshi Itoh¹

¹Laboratory of Information and Systems Engineering
Faculty of Science and Technology, Sophia University
7-1 Kioicho, Chiyoda-ku, 102-8554, Tokyo, Japan
{m-kamimu, r-kawaba, itohkiyo}@sophia.ac.jp
²Research and Development Headquarters, Yamatake Co., Ltd.,
1-12-2, Kawana, Fujisawa, 251-8522, Kanagawa, Japan
kumagai@atc.yamatake.co.jp

Abstract. To make collaboration tasks work effectively, it is very important to analyze what collaborators are doing, which position they are in, and how they communicate with each other. Multi-Context Maps (MCM) and Collaborative Linkage Maps (CLM) together constitute a solution for such demands. These methods analyze each collaborator's position precisely, so these models capture important information in the process. In this paper, we propose to extract E-R models, which are widely used to design databases in the early phases, from MCM and CLM. The process of composing E-R models still relies on the designers' experience or their intuition, which often causes inadequate formalization of complex requirements. Our extraction provides a systematic method of designing data models. We also propose how to apply the extracted model when designing databases.

1. Introduction

E-R models [1] are widely used in designing databases for commercial and engineering applications. An E-R model consists of *entities*, *relationships* and *attributes*. The process of specifying entities and relationships usually starts by gathering information by listing the nouns, which are the candidates of the entities, in the system. After you get the entities, the process of defining the relationships between the entities follows. This method is based on the concept of data-oriented analysis, which says that data is independent from the business processes and this makes the system more flexible when the business process changes. We think that this method can be useful when very precise analysis is needed, but the process can be time consuming, and also it does not have any criteria when to stop analyzing.

We think that it is necessary to focus the scope of the system to make the analyzing process effective. The STEP (STandard for the Exchange of Product Model Data) [2], an international standard for exchanging CAD data, suggests that the person who analyze the system should compose IDEF0 [3], a function model, to define the scope of the system. After the model in IDEF0 is composed, then IDEF1x [4] model, which is a data model, should be composed, based on the knowledge in IDEF0. The STEP suggestion shows us that mapping the things, which activities processes in the process

models, to the entities in the data model is reasonable because activities change the entities, which we usually need to manage by using information systems.

IDEF0 diagrams graphically represent the function of the system and can easily capture the information from domain experts effectively, but this easiness sometimes results in ambiguous specifications of the system [5]. This ambiguity makes the extraction of the data models a tough job.

We propose a method of extracting an E-R model from both Multi-Context Map (MCM) [6] and Collaborative Linkage Map (CLM) [6], which defines relationships among collaborators and resources based on the concept of Context Map. MCM has a precise syntax that keeps the model free from ambiguity, and CLM, which is the complementary of MCM, captures state transition of the entities. This process enables designers not only to capture important information and their relationships in the collaborating processes, but also to get criteria when to stop analyzing for getting a prototype of the data model of the system. The system designer can use this prototype as a starting point for discussion with the clients. We also have to say that our extraction does not get only the physical structure of the database, but also the “views” which provide the user new ways to use data that meet the organization’s need.

Multi-Context Map (MCM) is a method to systematically analyze collaboration tasks [7] among people and machines that perform tasks in the systems, and resources, such as information and materials, of complex systems. MCM defines *Context Map* as the minimum element, which represents the situation people or machines collaborate with each other, like exchanging information, asking someone to do the job, or so on. We call such people or machines, or organizations, *collaborators*. There are many kinds of such situations in the business process and we think that we can describe the whole process by the combinations of context maps, as the name, Multi-Context Map, shows.

Collaborative Linkage Map (CLM) [6] is a way to analyze state transitions of each personnel and resource. Those personnel does not exist alone but communicates with each other in the business process so each diagram is linked, based on the context map defined in MCM.

We have to say that we need both MCM and CLM to have an accurate description of the system. They are very closely related and using only one of them causes us to view only one aspect, which we think that is not enough.

In section 2, we discuss related works that takes near approaches.

In section 3 and 4, we introduce MCM and CLM, first about the concept and later about the legends of the graphical diagrams.

In section 5, we introduce E-R models, which are widely used in designing databases.

In section 6, we propose a guide of extracting E-R models from both MCM and CLM diagrams.

In section 7, we demonstrate the guide by applying it to a practical example of “Airport business, supporting one transfer”.

We conclude the paper by evaluating our guide and discussing issues of applying the extracted E-R models to next phase of design.

2. Related Works around MCM and CLM

MCM has a close relationship with IDEF0, a function model that describes the function of the system. A semi-automatic converter between MCM and IDEF0 is proposed [8]. IDEF0 is very flexible so it is powerful as communicating tools, but sometimes, where accurate analysis of collaborating processes is needed, this flexibility causes the diagram to get ambiguous. MCM has a more strict syntax than IDEF0 does to have an accurate description of the system. MCM has *tokens*, *information*, and *materials* to classify the type of the resources, while IDEF0 does not have such syntaxes. MCM also defines *junctions* to show the flow of the system explicitly, while IDEF0 does not. We will introduce the detailed syntax of MCM in the next section.

Petri Net [9] is a method to analyze state transition. It captures asynchronous, concurrent, and distributed systems, both conceptually and mathematically. However, when analyzing collaborating processes where many collaborators work in the process, the view of the whole system can become complicated and the designer cannot easily capture the whole view of state transitions in the system. CLM is a powerful alternative to analyze the state transition of the collaborating processes because it defines state transition diagrams for each collaborator and resource, and classifies all those states into three types, *enaction state*, *commission state*, and *dormant state* and links each diagram. This classification can make the analysis simpler. We will introduce CLM in more detail in section 4.

3. Multi-context Map

Multi-Context Map is a method based on the concept that business processes consist of collaboration of people or machine or anything that performs business tasks.

MCM defines *Context* as the most fundamental element. Context is the situation that one collaborator asks the other to do the intended job. The context consists of two *perspectives*, one from the requestor and the other from the performer. MCM defines *perspectives* as not only the opinion of each collaborator, but also a viewpoint, or a position from a person, an organization, or their combination. The *Context Map* represents the Context graphically. (See figure 1). The perspective on the left of the context is the *left-hand perspective*, who requests, and the one on the right, the *right-hand perspective*, who performs the requested job. The work usually flows from the left to the right.

We introduce the syntax about Multi-Context map briefly in this section. Multi-context map is the combination of context maps.

In figure 1, the passenger, who is in the left-hand perspective, asks the receptionist, who is in the right-hand perspective, to “check-in”. You can see that three kinds of arrow are coming from the left. One of them, in a broken line, says, “Passenger arrived”. This is *token*, which defines the fact understood between both perspectives. Token address the fact from the requestors and triggers the performer to start his jobs. In figure 1, the token, “Check-in completed” address the fact to the next context.

The arrow in the solid line defines *materials*, which represents physical things, which flow the system and receive service or make actions, and are usually tangible

and have weight, such as raw materials, facilities, tools, and includes people who receive service by collaborators.

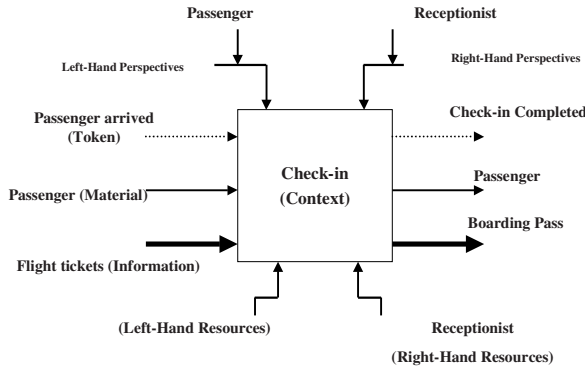


Fig. 1. A view of a Context Map

The arrow in the bold solid line defines the *information*. Information differs from token, as collaborators determine their status by token and use information for their jobs. Information in MCM gives or receives information to and from the collaborator. It is not tangible and does not have weights like materials. The examples of information are such as information on the charts, blueprints, etc.

There are two kinds of *resources* plugged in the bottom of the context. The resources Left-hand perspectives uses to do their jobs are the *left-hand resources*, while *right-hand resources* are right-hand perspective's resources. Resources are different from materials and information defined above. These resources appear only in the plugged context and do not flow the system, while material and information move around in the system. As we can see in figure 1, "Receptionist" as right-hand resources, means they do not only listen to the request, but also do the jobs. The receptionist is also a required resource, such as any other tools.

MCM has five kinds of junctions, "B", "Du", "De", "Sy" and "Se", to show the behavior of the workflow clearly. Figure 2 describes the MCM graphically. "B", "Du" and "De" has one input and multiple outputs, while "Sy" and "Se" have multiple inputs and one output. "B", which is represented in figure 2, is an abbreviation of "Branch" which defines the situation that one thing goes only to the one of the outputs and it depends on the condition where to go. "Du" stands for "duplication", which copies the input to all of the outputs. "De" comes from "decomposition" and the input decomposes, which means one thing breaks, into two or more outputs. "Sy" comes from "Synchronization" and waits all the supposed multiple inputs come before producing the output. "Se" is from "Serialization" and joins the multiple inputs to the output like a flow in the river.

In figure 3, we show the practical example of MCM, which describes the part view of the "airport business, supporting passengers including one transfer". The workflow starts from "Airport1" and move to the first arrival, "Airport2". The final destination

is “Airport3” in the model. The staffs named “Receptionist1” means “the receptionists at airport1”.

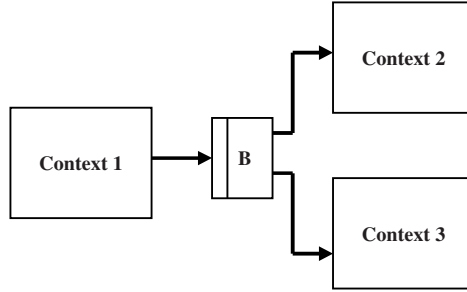


Fig. 2. Junctions in MCM (Branch Junction)

4. Collaborative Linkage Map

As we stated in Section 1, we need both Multi-Context map (MCM) and Collaborative Linkage Map (CLM) to represent the system properly and they are complements. We introduce Collaborative Linkage Map in this section, which is useful in analyzing state transitions.

Collaborative Linkage Map (CLM) defines *Personnel-State Transition Diagram (P-STD)* to represent the state transition of the collaborators and defines *Material-State Transition Diagram (M-STD)* to represent the state transition of materials or resources, as their components, and *Collaboration-State (C-S)* connects both of them based on the context defined in MCM. CLM also defines *Communication-Inventory (C-I)* to represent the information handed between P-STDs. An example of CLM is illustrated in Figure 3.

Collaboration-State does not exist alone. It always has connection with other states those come from the contexts in MCM.

Both P-STD and M-STD represent situations by means of states of three different kinds. *Enaction state* represents the situation that P-STD is acting on the M-STD and connects both STDs with Collaboration-State as illustrated in figure 4. For example, the “doctor”, as P-STD, writes the conditions of the patient to the “chart”, which is M-STD, then we define enaction states in each STD and the collaboration-states connect the “doctor” and the “chart”. *Commission-state* is defined when one P-STD has any kind of relations with other P-STDs and the information exchanged in such situation is defined as *Communication-Inventory*. When the “receptionist” hand the “chart” to the “doctor”, then the receptionist and the doctor has commission-state defined in their P-STD, and the chart becomes the communication-inventory. *Dormant state* is the state neither of them and does not have any relationships to other state diagrams.

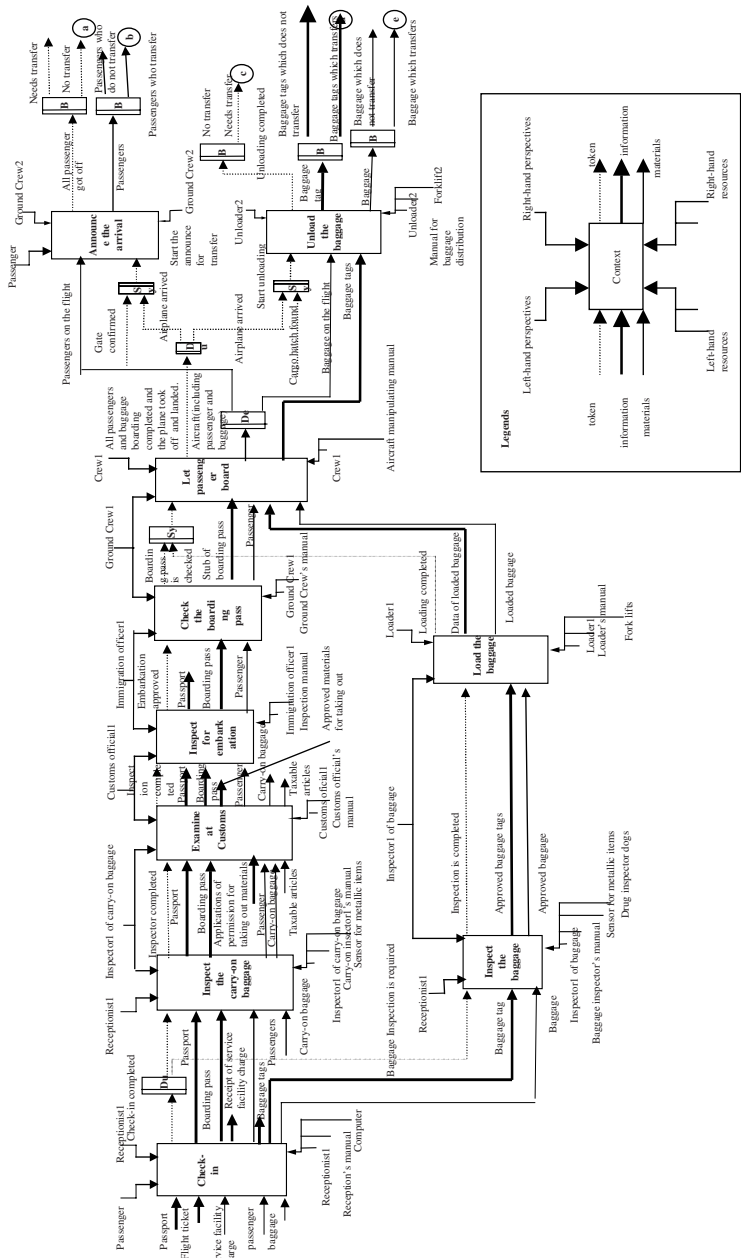


Fig. 3. A part view of MCM of the airport business supporting passengers and management of baggage

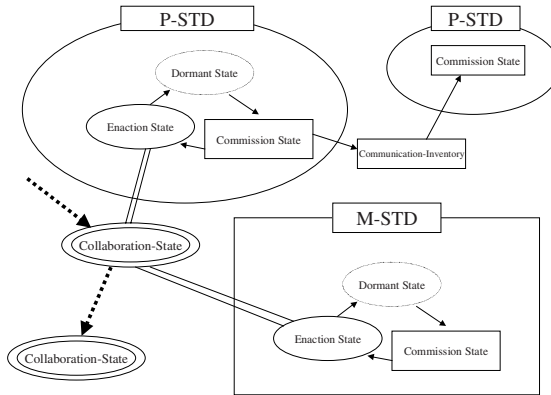


Fig. 4. A view of Collaborative Linkage Map (CLM)

We show a practical example of CLM, also representing the “Airport business, supporting passengers and the management of their baggage at international flights including one transfer” in figure 5. Figure 5 contains only one part of the whole system because of the space of paper.

5. E-R Models

The E-R (Entity-Relationship) model [1] is based on a perception of a real world, which consists of a set of basic objects called entities and relationships among these objects. It is developed to facilitate database design.

5.1 Entities

An *entity* is a set of an object that exists and is distinguishable from other objects that are concrete or abstract of the same type. The set of all persons having an account at a bank, for example, can be defined as the entity customer. Similarly, the entity account might represent the set of all accounts in a particular bank.

An entity is represented by a set of *attributes*. Possible attributes of the customer entity set are name, social security, street, and city.

5.2 Relationships

A *relationship* is an association among several entities. For example, we may define a relationship that associates customer “Harris” with account 401. This specifies that Harris is a customer with bank account number 401.

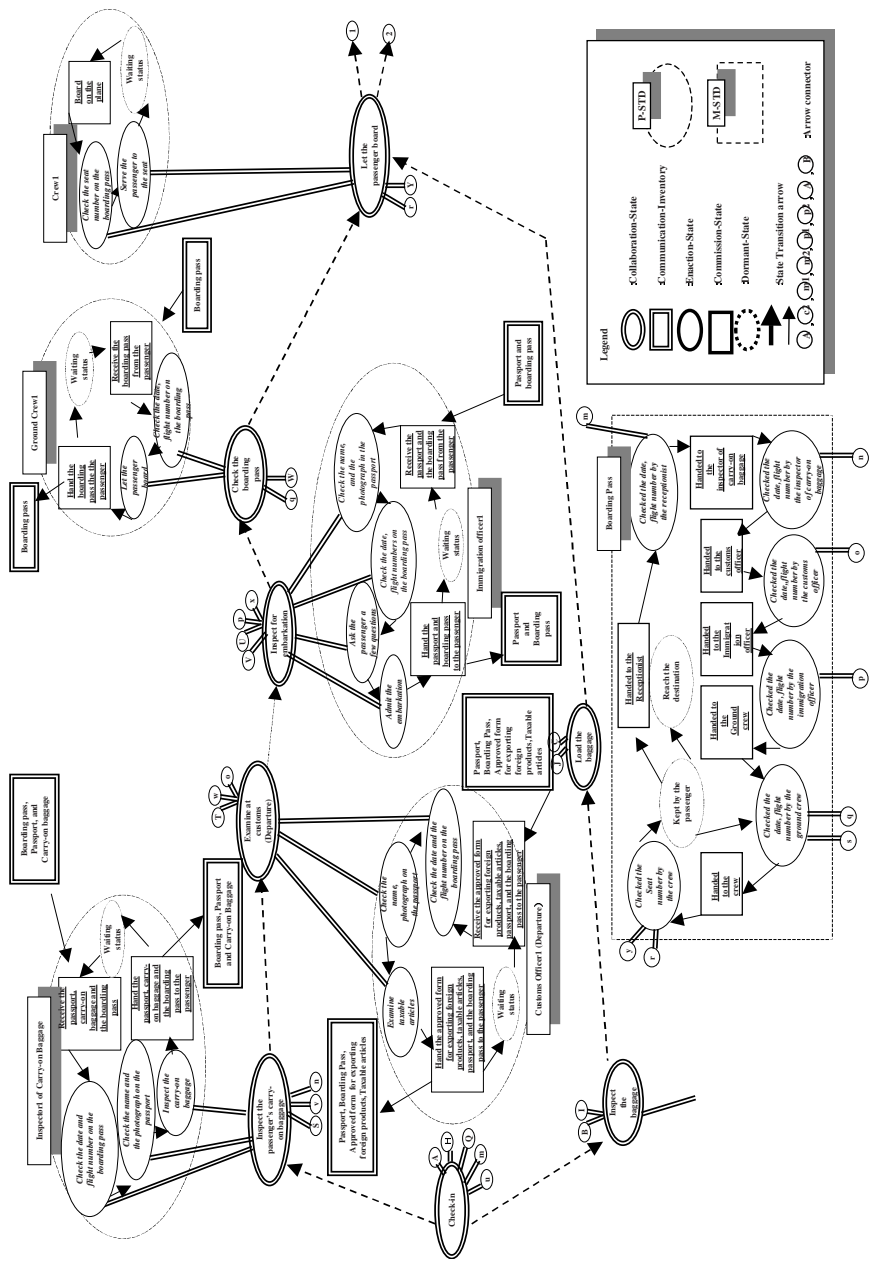


Fig. 5. A part view of CLM of the airport business supporting passengers and management of baggage

5.3 E-R Diagram

E-R models are described graphically by an E-R diagram, which consists of the following components:

1. Rectangles, which represent entities.
 2. Ellipses, which represent attributes.
 3. Diamonds, which represent relationships.
 4. Lines, which link attributes to entities and entities to relationships
- We show an example in figure 6.

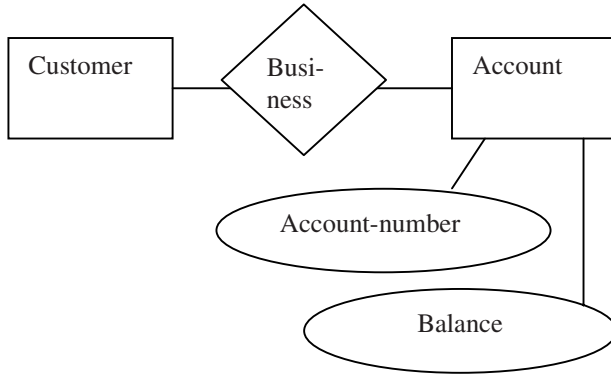


Fig. 6. E-R diagram

6. Process of Extracting E-R Models from MCM and CLM

We propose to extract an E-R model from MCM and CLM. CLM defines not only the state transition of its components alone but also links them by the notion of collaboration-states, which comes from contexts in MCM. This fact shows us that MCM and CLM define not only relationships between entities but also the detailed information between the relationships defined in the enactment states. We propose to extract an E-R model that contains not only the physical schema candidates but also *views* [1], which enable the users to manipulate data more effectively. A *view* is a virtual table, which is defined in relational databases, and provides the user with an alternative representation of the data. It looks like a physical data but it has no data on its own. It is instead the result of an SQL SELECT statement. We discuss this issue in the next section in details.

The process is not automatic and it needs the designer's decisions to determine which items are entities and which are attributes.

The steps follow as below. Step 1 to 3 is a brief description for composing MCM and CLM. Our method starts from step 4. See notes for examples, which defines the case for extraction from MCM and CLM, "Supporting system for international flights" shown in the previous section.

We show a brief example of the steps in figure 7.

(Step 1) Define MCM. Please consult [2] for details.

(Step 2) Define a sketch or outline of CLM by using MCM defined in step 1. In this step, capture the overall structure and do not have to define the STDs in detail.

(Step 3) Define the CLM, especially the inside of each STD. We recommend writing as concrete as possible, especially the movement around the information, such as "The information checked", or "Information added".

(Step 4) Map the P-STD in the CLM to entities. We think that the collaborators in the system are important entities. The collaborators make action in the process and have close relationships to the data around them.

(Step 5) Pick up the M-STD in the CLM as candidates of entities. These entities are the most fundamental part of the system and the system manages this kind of information. We usually map all of them as entities but in some cases, information in CLM is defined to manage materials. We are now designing a database to manage information and not caring about the physical side of the material. Therefore, we can omit those materials if the information represents the vital information about the materials. For example, in a hospital, "Chart" plays the role to manage information for each patient. Therefore, we do not have to track both "Chart" and "Patient" to manage information about patients because "Chart" usually contains all of the information for each "Patient". In such cases, we only map "Chart" to the entity and omit "Patient".

(Step 6) Take notice to the enaction states of M-STDs. If there are any kind of description related to the movement around information, such as "Information checked by whom" or "Information written or added by whom", then define an entity dependent to the entity mapped from the M-STD that contains the enaction state and define a relationship "includes" between those entities. See note 1

(Step 7) Map the collaboration-state to the relationship, which connects the entity defined in step 4 and step 6. See note 2.

(Step 8) Define the attributes of the entities defined at step 5 from information extracted at step 6. See note 3.

(Step 9) Define the relationships between entities mapped from M-STD in step 5. This step requires the designer's judgment. See note 4.

(Step 10) Exclude the entities that have no relationships with any other entities.

(Step 11) Define the attributes of the entities mapped from P-STD in step 4. The designer also has to add his information to define them.

(Step 12) Define the attributes of the entities mapped from M-STD in step 5. This step also requires the designer's judgment.

(Step 13) If still the diagram is not good and need more information in details, return to step 3 and add more information to the original CLM.

<Note 1>

Suppose that there is an enaction state that say, "Checked the name and the photograph by the immigration officer1" in the M-STD "Passport". In this case, define a dependent entity "Information check by the immigration officer1 on the passport" and relationship between this entity and the entity "Passport" which is previously mapped from M-STD. This new entity has two attributes, "Name" and the "Photograph".

<Note 2>

In the case for collaboration-state "Inspect for embarkation" two STDs are connected, one is P-STD "Immigration officer1" and the other is M-STD "Boarding Pass". Map the collaboration-state to the relationships between the entity "Immigration officer1" and the "Information checked by immigration officer1 on the boarding pass" because the entity defined from enaction state has more closely related information in the collaboration-state. Do not relate directly to the entity "Passport".

<Note 3>

In the M-STD, "Passport", an enaction state is defined. It is "Name and Photograph is checked by the Immigration officer1". In this case, define "Name" and "Photograph" as attributes of the entity "Information checked by the Immigration Officer1" that is a dependent entity to the Passport.

<Note 4>

In the supporting system for international flights as we shown for practical example in MCM and CLM, Passenger, Baggage, Passport, and the Boarding Pass come simultaneously and we can think that the Passenger usually holds all of them. We define relationship "Own" between "Passenger" and the rest of them.

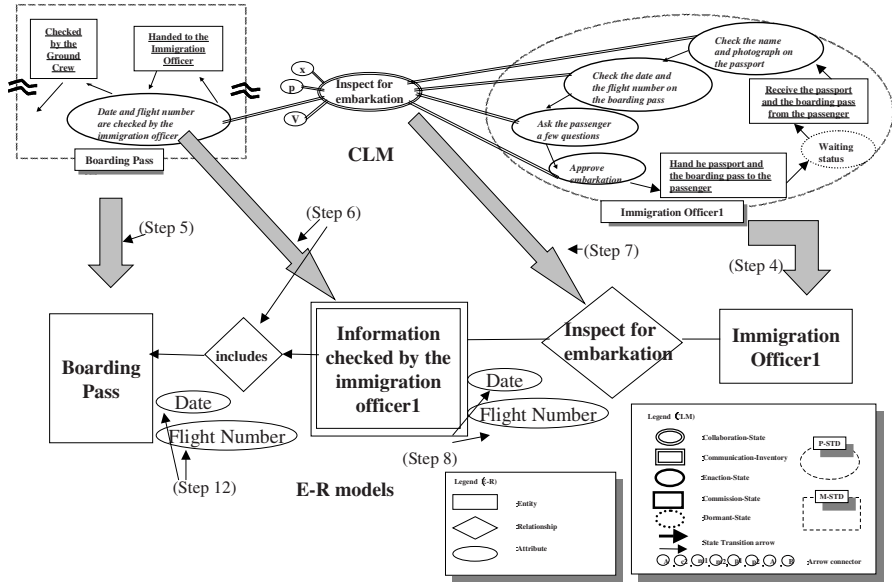


Fig. 7. Process of extracting E-R models from MCM and CLM

7. A Practical Example of the Extraction

We show the whole view of the extracted model of the “System, which supports the passenger, and the management of their baggage at international flights, including one transfer”. In this system, the process starts from the arrival of the passenger at airport, which we call airport1. The process goes from the check-in, embarkation, inspection, and arrival to the second airport, which we call airport2. The passenger transfers at this airport and has another flight to the final destination airport3. Figure 3 shows the part views of the MCM of this process and figure 5 the part view of the CLM of the same process. In figure 8, we show the whole view of the extracted E-R model.

This model also contains the dependent entities. We think that in our method, the dependent entities can become views, as we mentioned in the previous section, because this dependent entities does not contain any other information while the other independent entity contains. All information in these dependent entities comes from the entities defined from M-STD. These dependent entities become views in the relational databases by using the SQL SELECT statement. The view represents the information used by each collaborator and help extracting transactions.

In figure 9, we show an example of the E-R model, which does not contain the dependent entities. We believe that the model, which defines the real table, will be like in figure 9. Erasing the dependent entities and making the relationships direct between entities defined from P-STD compose this model.

8. Conclusion

When designing databases, specifying entities and relationships is usually done manually, and important information cannot be captured.

Our method extracts not only the information for physical schema designs of the database but also the views, which is also a critical point in the physical design of the databases. Views have many potential benefits, and provide the user new ways to use data that meet the organization's need, but designing the views are as hard as designing the physical structure of the database. The dependent entity defined in figure 8, shows the information each user, usually the collaborator defined as P-STD. We believe that our extracting process is also useful to design the views more effectively, and shorten the process for getting the prototypes.

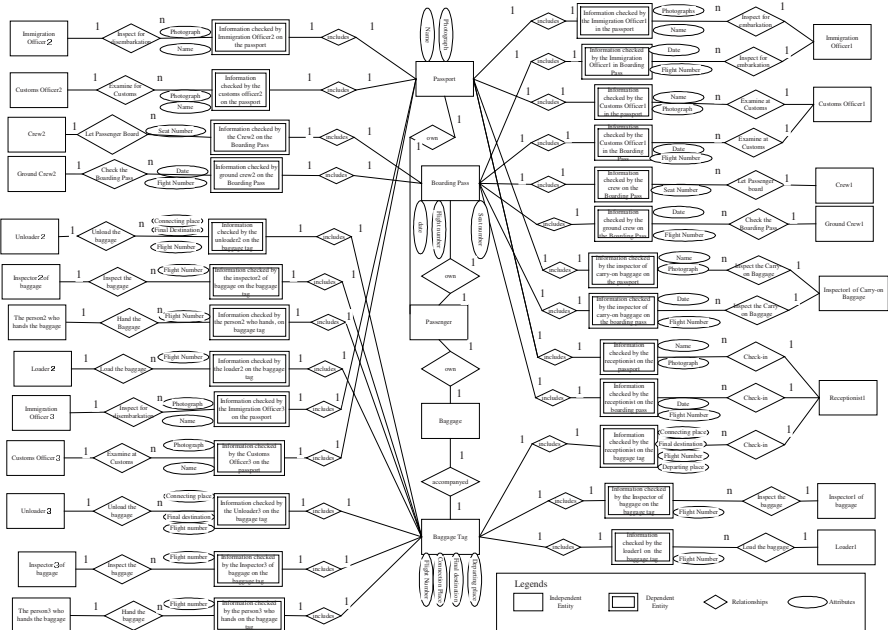


Fig. 8. System which supports passengers and management of their baggage at International Flights, extracted from MCM and CLM

Acknowledgement. We really appreciate to Ms. Akiko Hasegawa who gave us useful advice all the time. She is one of the members who invented the MCM and CLM.

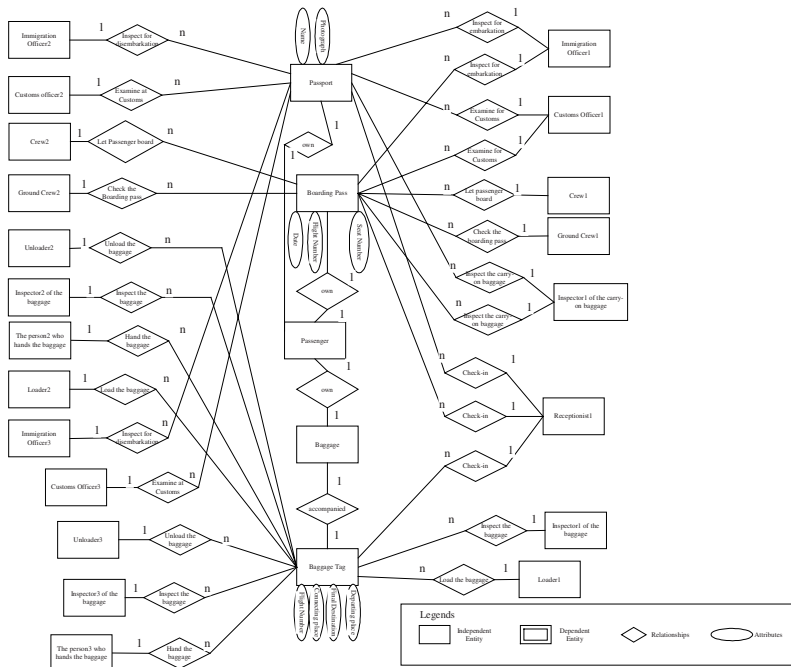


Fig. 9. E-R model which represents the physical database of the System, extracted from the model in figure 8

References

1. R. K. Stephens and R. R. Plew: Database Design, Sams publishing (2001)
2. Fowler, Julian: STEP for Data Management, Technology Appraisals. (1995)
3. D.A. Marca, C.L. MacGowan: IDEF0/SADT Business Process and Enterprise Modeling, Eclectic Solutions Corp. (1988)
4. T.A. Bruce: Designing Quality Databases with IDEF1x Information Models, Dorset House Pub. (1992).
5. H. Kitagawa, T.L. Kunii.: Comparative Evaluation of Software Specification, bit, Vol.10, No.10, pp.1160-1191, (1978), in Japanese.
6. A. Hasegawa, S. Kumagai and K. Itoh: Collaboration Task Analysis by Identifying Multi-Context and Collaborative Linkage, CERA Journal, Vol.8, No.1, pp.100-108, (2000)
7. S. Kumagai: A study on Analysis Methods in Collaboration Domain, Doctoral Thesis, Sophia University, Tokyo (2000)
8. K. Inoue, S. Kumagai, R. Kawabata, and K. Itoh: Enhancement of Collaboration Process by Bi-directional Converter between IDEF0 and Multi-Context Map, FOSE (Foundation of Software Engineering) '99 p.260-267, (1999) in Japanese
9. W. Reisig, G. Rozenberg: Lectures on Petri Nets I: Basic Models Springer (1998)

Living Manufacturing Systems with Living Organizations

Noriaki Kurosu¹ and Shunji Yamada²

¹Toyota Motor Corporation, Information Technology & Engineering Division
7, Teho, Teho-cho, Toyota, Aichi, Japan
kurosu@mail.toyota.co.jp

²Yamatake Corporation, Research and Development Headquarters
2-12-19, Shibuya, Shibuya-ku, Tokyo, Japan
syamada@pres.yamatake.co.jp

Abstract. Borrowing the title of Nobel-prize laureate physicist Erwin Schrodinger's work "What is Life?"[1], we ask "What are Living Systems?" This paper attempts to answer the question by looking at a manufacturing information system and an organization model as examples. The fundamental concept is "harmonized autonomous decentralized systems." We will introduce examples that show how this concept is reflected on manufacturing information systems, where humans and machines coexist and work together in the production line, and identify the characteristics of the concept. We will show that, with the current level of technology, human-machine coexistence systems are needed to meet the requirements for manufacturing systems, especially in order to respond quickly to environmental changes. We will also evaluate psychological aspects such as challenges and satisfaction of people involved in operating the systems. Finally, we will introduce issues remaining yet to be addressed.

1 Introduction

It is difficult to answer the question "What are Living Systems?" posed in the abstract. Nevertheless, it may be possible to answer the question, since if we consider the systems merely created by humans like manufacturing systems, they are still simple despite of their wide range of coverage compared to biologically living systems existing in the natural world. Though it may be incomplete, Table 1 shows some key words of "Living Systems" as a part of the answer, along with the requirements for manufacturing systems derived from them. These may also serve as the key words for representing the final goal of our HUMACS (HUMAN-MACHINE Co-existing Systems) study.

Table 1. Keywords of Living Systems and Requirements for Manufacturing Systems

Keywords of Living Systems		Requirements for Manufacturing Systems
- Self-organization	- Intelligent resonance	- Developability
- Self-proliferation	- Symbiosis	- Flexibility
- Self-recovery	- Co-existence	- Re-configurability
- Autopoiesis	- Catastrophe	- Improvability
- Metabolism	- Fluctuation	- Maintainability
- Co-evolution	- Metamorphic	- Accessibility
- Co-creation	- Synergism	- Comfortability
- Genetic mutation	- Entertainment	
- Dissipative structure		

These requirements are obtained from the key words translated in the context of manufacturing systems so that we can respond quickly to the dynamically changing manufacturing environment shown in Fig.1. We abstracted these terms from useful aspects in human social systems as well as from the history of biological evolution that allowed the survival of living things that could adapt to environmental changes.

Among the items in Fig.1, those concerning human factors are increasingly emphasized lately. This is because we are beginning to realize criticality to mobilize the collection of broad wisdom of highly skilled humans in order to respond quickly to complicated events, as is seen in the globally expanded business operation in today's age of large-scale competition often dubbed "mega-competition."

We often hear expressions such as "the 21st century is the age of knowledge and information society," and indeed rapid progress in IT is creating a borderless, globalized world. The universal availability of knowledge, symbolized by the key word "ubiquitous," is progressing quickly. This paper suggests how we should address and are addressing these issues based on the fundamental concept of "autonomous."

☐ Market Change	☐ Labor Environment
▪ Diversifying needs	▪ Attractive workplace
Emphasis on affluence	Self-actualization/Sense of achievement
Emphasis on individuality	▪ Demand for higher academic qualifications and advanced skills
Emphasis on sensitivity	▪ Education, training and human development
▪ Higher quality of life	
▪ Shorter product life cycle and development time	☐ Increasingly globalized and borderless world
▪ Shorter delivery time	☐ More information-intensive, networked environment
▪ Increased cost competitiveness	☐ Global environmental problems

Fig. 1. Changes in Manufacturing Environment "Quick Response to Changes"

2 Concept of System Construction – Harmonized Autonomous Decentralized Systems –

2.1 Definition of Terms

As shown in Fig.1, manufacturing systems must solve complicated and diverse problems in terms of space (globally) and time (24 hours-a-day and/or immediately). There are challenges in many areas, and "autonomization" may be the only way to solve them. This is the concept that all surviving creatures have learned from the wisdom of nature through a long period of trial and evolutionary history.

A top-down, hierarchical and rigid large-scale system cannot work on problems quickly and flexibly. Without utilizing the functions of “key words of living systems,” such as harmonization of reasonably decentralized modules, co-evolution and co-creation, in Table 1, problems will remain unsolved. This paper calls the system with such functions an “autonomous decentralized system.” This term is a compound of words: “autonomous” as in “autonomous nervous system” and the system that is defined as “many interacting components working together toward one goal.”

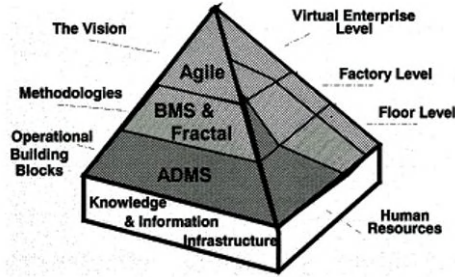


Fig. 2. Cooperative and Integrated Manufacturing Systems in NGMS

Thus, an “autonomous decentralized system” is defined as “a system where each individual component acting based on its own rules achieves its goal in cooperation with other components.” Sometimes such a system is called a “cooperative autonomous decentralized system” or a “harmonized autonomous decentralized system” to emphasize the aspect of cooperation, collaboration and harmony with others. Toyota adopts the expression “harmonized autonomous decentralized system” in this regard. The prefix “harmonized” was chosen with the image of a controlled movement of centipede’s legs, a boat race (the eight) and an orchestral performance in mind as you can see in Fig.3.

The concepts, such as decentralization, cooperation, harmony and autonomy that are based on the key words listed in Table 1, are widely studied. Under the IMS (Intelligent Manufacturing Systems) program, the next-generation manufacturing systems project (NGMS) proposes such autonomous decentralized systems (ADMS) as described in this paper, biological manufacturing systems (BMS), and Fractals[2]. The holonic manufacturing systems project (HMS), on the other hand, suggests the concept of “Holon” presented by A. Koestler that features on the idea: “a part is a whole and an individual particle at the same time.” [3] Fig.2 shows the conceptual diagram of the cooperative and integrated manufacturing systems in NGMS, while Fig.3 shows the conceptual model for manufacturing systems based on living things. The common theme is to “learn from the flexibility fostered during a long history of living things.” There are no universal norms existing beyond time and space by nature due to the limitations and diversity of humans and machinery systems. The common idea is to rather accept their multi-dimensional existence, thereby analyzing, studying, and suggesting their evolutionary, interactive, or collaborative aspects. We attempt to quickly and flexibly respond to the environment surrounding manufacturing through

the use of this approach. Although those systems mentioned above are all similar from a macroscopic viewpoint in terms of purposes, concepts, and models, the term “harmonized autonomous decentralized system” is adopted in this paper.

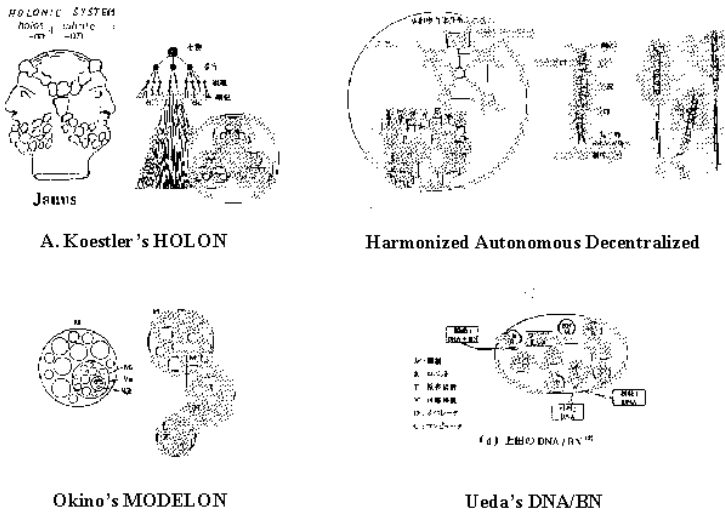


Fig. 3. Conceptual Model for Manufacturing Systems

2.2 Why Autonomous Decentralized System?

There is a limit to the inherent abilities and performances of human beings as living things. Meanwhile, machines have no capacity to respond to unexpected changes (variability); they are practically powerless. Human beings have learned to achieve goals by organizing. (Why do organizations exist? People organize because they believe that it is the most efficient way to reach their goals. [4]) Machines too are trying to address the problem by combining appropriately sized functional modules suitable for the environment at the time as well as by integrating and systematizing them through networking. The ones that cannot keep pace with the environmental changes will be unavoidably cut off or replaced by new modules. In this way we attempt to respond to environmental changes quickly and flexibly. Sometimes a large-scale hierarchical system is more efficient. However, for systems having the requirements of “Living Systems” as shown in Table 1 together with the current level of machine and system technology, we have no choice but to adapt the structure of the autonomous decentralized system presented in this paper.

3 Human Factors

3.1 Definition and Relationship with Manufacturing Systems

Not only from the technological and economic perspectives but also from the viewpoint of manufacturing lifecycle activities including maintenance and disposal of facilities, it is impossible to build a fully unmanned manufacturing system. We have learned from the experience in a Japanese national project called “Unmanned Factory” and the pursuit of building an automatic assembly line for the automotive industry during the years of the bubble economy that manufacturing systems will require a human-centered system structure, with human beings as the backbone module, for many years to come. We need to promote research on human factors in order to build and operate systems using human power more effectively. What do “human factors” mean when we say “Human Factors in Manufacturing Systems?” Following “Human Factors in Flight”[5], we define them in this paper as follows: “Human factors are about people. It is about people in their working and living environments. It is about their relationships with machines and equipment, with procedures and with the environment about them. And it is also about their relationships with other people.” Fig.4 shows how human factors and manufacturing systems relate to each other.

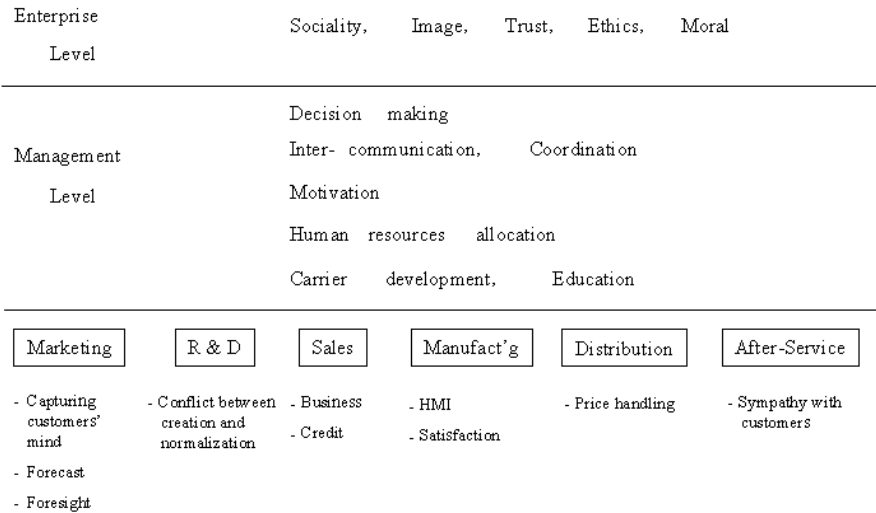


Fig. 4. Relationship between Human Factors and Manufacturing Systems

3.2 Human Factors Requirements

To satisfy the multitude of requirements, manufacturing systems such as automotive assembly lines must be human-centered systems as we described earlier. Thus, first and foremost the workplace must be safe and the job has to be challenging and deliver

Table 2. Users' Requirements Based on Human Factors

Major item	Intermediate item		Index
Safety	Safe working environment	Machines and systems do not go out of control or break down.	Safety Dependable Reliable, Robust Fault-tolerant Fault-avoidance Fail-soft Fail-safe
		In case they break down, they do so safely.	
		Users can understand machines and systems. (Concept, functionality, and performance)	Accountability Visible Legible Transparent
		Easy to use Easy to train Easy to improve Easy to maintain Monitorable and traceable	Accessibility Usability Trainability Simplicity Improvability Evolvability Maintainability Repairability Renewability Monitorability Traceability
	Easy and reliable safety management, supervision, and diagnosis		Monitorability
Working environment	Prevention of work-related diseases	- Comfortable working environment Temperature, illumination, sound, vibrations, poisons, radioactive rays, lasers, high voltages - Liberation from heavy muscular labor - Liberation from unskilled labor - Moderately sophisticated facilities and systems	Amenity
	Maintenance of the will to work		Adaptability Adequatability Accessibility Affinity
Labor management environment	Working systems to raise Motivation Participation Quality of working life Well-being Promotion of team-work Reducing health-care	- Intentions of others (managers, planners, supervisors, co-workers, users) can be grasped. - Workers can easily access necessary information. - Workers can utilize their experience. - Workers can get cooperation. - Workers can expect to have their proposals and ideas adopted. - Workers can develop themselves. - Workers can feel they are trusted and expect to do their jobs on their responsibility. - Workers can expect to have their performance, improved skills, and results recognized by bosses, co-workers, organization, and society.	Accountability Visible Legible Transparent Open Accessibility Affinity Improvability Extendibility Developability Autopoiesis Evolvability
	Challenging jobs and a sense of achievement - Job satisfaction - Accomplishment - Fulfillment - Achievement - Self-actualization Self-managing Self-expression Fair evaluation		Open Recognition Evaluation

a sense of achievement to workers. Table 2 shows such requirements for manufacturing systems based on human factors, while Table 3 summarizes the evaluation index for manufacturing systems listed in alphabetical order, derived from Fig.1, Table 1, and Table 2.

Table 3. Evaluation Index of Living Systems

acceptability	dependability	operability	reusability	stability
accessibility	developability	packageability	reversibility	suitability
accountability	disassemblability	portability	robustness	supportability
adaptability	discriminatory	practicality	safety	
affinity	disposability	profitability	scalability	
amenity	divertability	reachability	security	
associativity	durability	readability	serviceability	
autopoiesis	evolvability	recollectability	separability	
availability	expandability	reconfigurability	similarity	
capability	extensibility	recyclability	simplicity	tolerance
changeability	flexibility	reduceability	self-adaptability	traceability
compatibility	improveability	redundancy	self-generatability	trainability
complexity	interchangeability	reliability	self-organizability	transportability
consistency	interoperability	renewability	self-proliferency	transparency
controllability	legibility	repairability	self-sufficiency	usability
connectivity	maintainability	repeatability	self-renewing	variability
cooperatability	manipulability	replaceability	self-replaceability	versatility
coordinatability	mobility	resilency	self-reparability	viability
	modulability	restructuability	self-recovery	visibility
		returnability		

4 Example of Manufacturing Information Systems

4.1 Storage Line Control System between Painting and Assembly Processes[6]

We will present the manufacturing information system used to control the production sequence between the painting and assembly processes in an automotive plant as an example of a human-machine coexisting system and harmonized autonomous decentralized system. The purpose of this system is to decide the order of moving vehicles from the painting process to the assembly process. As vehicles flow from the body production process in the early production stream through the welding and assembly process, disruption in processing order or load fluctuation is caused by unforeseeable disturbances like mechanical problems or faulty processing. Similarly during the assembly process, various unforeseeable disturbances result in missing parts and other problems. Fluctuation of workload may also be caused in assembly line operation. It is scheduler that has responsibility to perform real-time replanning while taking into account possible missing parts in down stream and leveling of

workload. This procedure requires expert knowledge and experience. Also, much real-time information acquisition needed for the planning.

To cope with achieving a hard-to-predict task as mentioned above, most scheduling methods actually being used in manufacturing fields adopt heuristic algorithms. It is customary that solution is sought through mathematical programming for determinant processes, through simulation for stochastic processes, or through such heuristics for field know-how as AI(Artificial Intelligence), Fuzzy, and neuro.

In an autonomous module structure shown in Fig.5, mathematical programming is applied to ALC(Assembly Line Control) that is responsible for production command on the whole assembly factory, while production sequence control is carried out by application of AI, specifically an expert system being adopted. High flexibility and information processing ability that human-being has is utilized for planning and abnormal situation management, while on the other hand fast control ability that belongs to PLCs (sequencers) is applied to mechanical control of conveyor

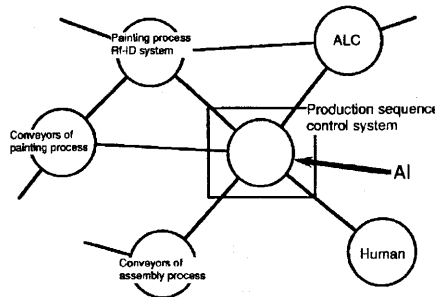


Fig. 5. Network of Autonomous Systems

ways, resulting in optimally combined systemization with work partitioning between humans and machines(PLCs, or computers) taking into account respective features and capabilities. This type of systemization would be inevitable for sophisticated systems where information/control and humans/machines are combined in a complicated way.

Fig. 5 illustrates an autonomous distributed system structure composed of conveyor control units, schedulers, and humans. Its line structure and AI-applied scheduler with an expert system are shown in Fig.6. Conveyor control that is to be carried out in milli-seconds with real time scheduling to be performed in a second processing about 60 rules has proved to successfully attain our objective satisfying economic requirement as well. The system has sufficient agility and flexibility to environmental changes like demand fluctuation by partial modification of autonomous modules, thereby demonstrating an example of a “Living Production System” satisfying requirements posted in Table 1 and Fig. 1.

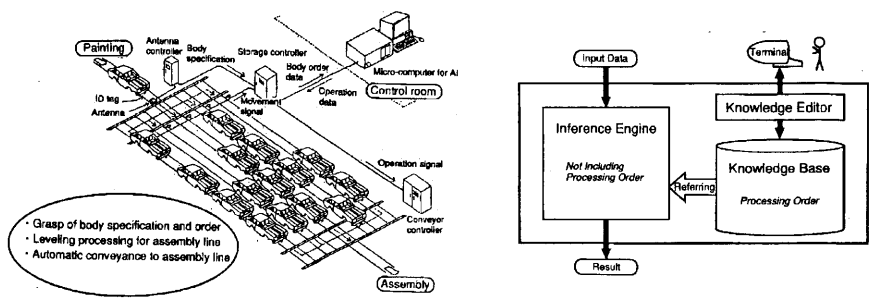


Fig. 6. Production Sequence Control System and Structure of Expert System

4.2 Evaluation from the View Point of Human Factors

Table 4 shows a result of polls taken in automobile industries in Japan on a question "In what occasion do you feel willingness for your job? ", and a list of selected key words from F. Herzberg[7], an American industrial psychologist well known for "Two-Factors Theory/Satisfaction-Dissatisfaction/Motivation-Hygiene". There are such many common key words found in relation to human factors between these two as "autonomy", "achievement", "opportunity to develop ones own special ability", "advancement: upgrading the skill and personal growth", and "recognition".

Table 4. Japan-US Comparison on "What do workers feel motivated?"

Case from a Japanese automotive company	Case from F. Herzberg
- Allowed to do the job on one's own responsibility - Feeling of achievement of the job - Accomplishment - Working together with co-workers - Self-improvement through one's job - Recognition by boss and co-workers of one's job performance	- Autonomy, Responsibility - Achievement, Accomplishment - Opportunity, Full use own skills - The work itself, Work value - Advancement, Challenging work to do - Recognition

In the case of the storage line control system mentioned earlier, it is left to veteran workers' duty to specify rules for schedulers and expert systems. Though being inferior to younger workers in muscular strength, they provide computer with

subject matter. At the signal entrance, we specified the stimuli to prevent the system from becoming obsolete, deteriorated or decayed. We placed a filter before the processor, and specified the control and management function, thereby selection of a processing mode being carried out based on experience and learning. These are shown in balloons in Fig.7. Activities such as Plan-Do-Check-Action (PDCA) in QC circles can be described using this model.

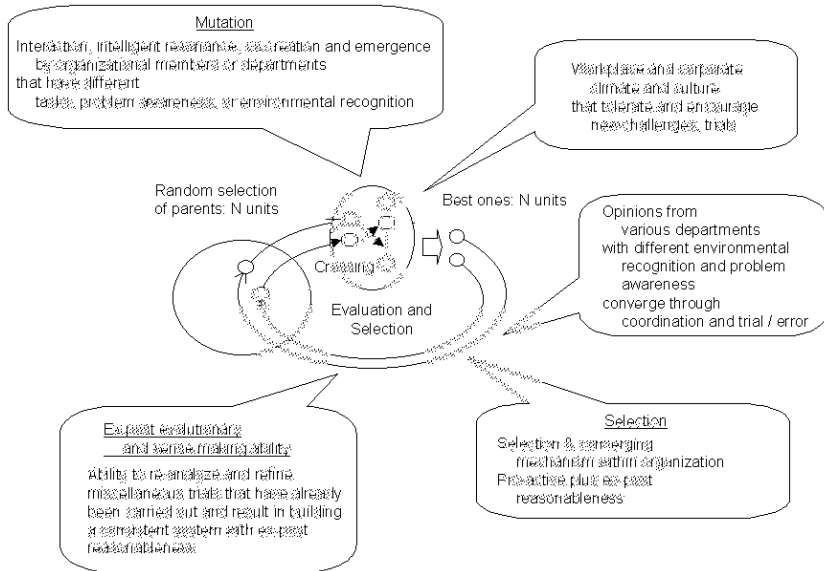


Fig. 8. Basic Model of Processor

Fig.8 shows a basic model of a processor. It emphasizes the images of mutation and evolution rather than logic and processing procedures to present a genetic algorithm model. This model clearly defines the members of the organization, rearrangement of procedures, evaluation by trial of new methods, selection, establishment, standardization, and transference. This model aims to help the motivated members of QC circles or similar groups to describe the improvement of the group's performance and members' motivation by using their ingenuity and creativity. These are shown in balloons in Fig.8.

Fig.9 shows an organizational model combining these basic models. Modules in each layer are connected via various networks based on a hierarchical structure and also connected with related modules in different layers in the hierarchy. This is a model case of a harmonized autonomous decentralized system achieving a mission by interrelated modules being organically connected.

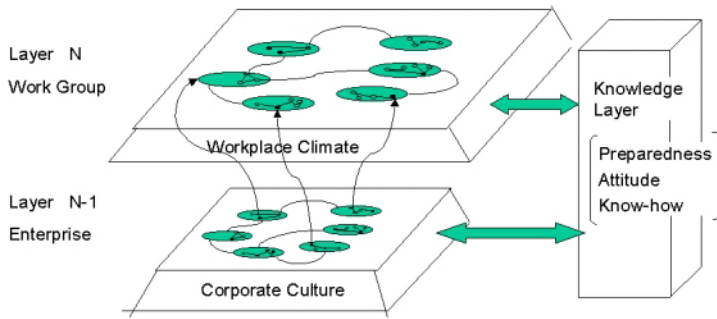


Fig. 9. Conceptual Model for Manufacturing

Fig.9 uses the terms employed in workplace in manufacturing. To solve problems, related personnel and organizations link together organically and produce results. The outcome and its process are evaluated and accumulated within people's minds and hearts as well as in organizations as knowledge and wisdom. These in turn lead to the creation of satisfaction and motivation in view of human factors aspects, while the same accumulated knowledge and wisdom lead to the creation of culture in organizations and societies. This is a model that describes new activities, improvements, creations, and innovations being carried out based upon a manner mentioned above. This is an example where the concept of an autonomous decentralized system is realized in a corporate workplace organization having its key idea taken from living beings.

6 Conclusion

We have used a manufacturing information system and corporate workplace organization to introduce the concept of harmonized autonomous decentralized systems, focusing on human factors. When we analyze in terms of a structural model "Living Systems" with vigorous people, organizations and corporations, those systems seem to form an autonomous decentralized system as introduced in this paper. This is perhaps not surprising because autonomous decentralized systems are based on living beings and "Living" is their key concept. Using examples, we have demonstrated that this concept is useful when we build and operate a manufacturing system that allows us to quickly and flexibly respond to environmental changes. We have found that it is necessary to increase the power of people, who constitute a pivotal part of the system. Multifaceted research including research on psychological aspects such as satisfaction and motivation will be indispensable to accelerate raising of human power. Research on modeling as described in this paper, especially research on a formula model that describes how wisdom is formed, accumulated and transferred within a system, is very important. Research on enterprise ontology conducted by PSIM (the European module of HUMACS), and that on self-organization and emergence currently studied in the IMS program and others will provide useful information.

Acknowledgements. We acknowledge that this research was conducted under the IMS International Joint Study Program and thank the people involved.

References

1. Erwin Schrodinger: "What is Life?" – The Physical Aspect of the Living Cell, Cambridge University Press (1944)
2. IMS International Joint Study Program, IMS0002, "Results of Research on Nest Generation Manufacturing Systems," MSTC/IMS Center (2001)
3. Arthur Koestler: The Ghost in the Machine, Hutchinson & Co., Ltd. (1967)
4. T. R. Mitchell, J. R. Larson Jr.: People in Organizations(3ed.), McGraw Hill (1987), pp.173-174
5. Frank H. Hawkins: Human Factors in Flight, Ashgate Publishing (1987), p.20
6. T. Ochiai, Noriaki Kurosu: "Development of Expert System for Production Sequence Control", Japan/USA Symposium on Flexible Automation, Vol. 2 (1992), pp.1433-1440
7. F. Herzberg: Work and the nature of man, World Publishing Co. (1966)

A Study on Human-Centric Real-Time Scheduling for PWB Assembly Line

Machiko Chikano¹, Yoshitaka Tomita¹, Yasuhiko Hiraide¹, and Eiji Arai²

¹ Yamatake Corporation, Research and Development Headquarters, 1-12-2 Kawana, Fujisawa, Kanagawa 251-8522, Japan

{machi, tomita}@pec.yamatake.co.jp, hiraide@atc.yamatake.co.jp
<http://www.yamatake.co.jp/>

² Osaka University, Department of Manufacturing Science, Graduate School of Engineering, 2-1, Yamada-oka, Suita, Osaka 565-0871, Japan

arai@mapse.eng.osaka-u.ac.jp
<http://www6/mapse.eng.osaka-u.ac.jp/index.shtml>

Aiming at promoting both a mass customized production, and higher motivations of human operators, we have conducted an analysis, and a simulation on the Printed Wiring Board (PWB) assembly line. Furthermore, we have developed a simulation model to realize our aim of the research in actual manufacturing fields.

Our simulation models contains “action-owners,” such as human operators, machines, robots, and so forth, and also “unit-processes” composed of one or more actions and independent from any specific action-owners. A real-time scheduling system employed such kind of models can visualize dynamic exchanges of action-owners between processes aligned with situation changes, and consequently contribute to realize a mass customized production in a human-machine cooperative manufacturing environment.

1 Introduction

In recent years, many manufacturing industries have been exposed to global competition. They have been forced to take various measures to survive in the market by responding to diversifying needs, and delivering products to customers on shorter lead times.

The electronic device manufacturing industry is a typical example. Our company, one of such a manufacturing firms, has been making continuous efforts to realize a practicable system for the production of a wide variety of models in small quantities each, and also varied models in varied quantities each. We have achieved some results, especially in final assembly that depends primarily on manual work. However, we have achieved little progress on our PWB assembly lines, in which we have to depend on machinery to deal with the rapid miniaturization of parts. This is because the existing surface mounting equipment is designed for mass production, featuring high speed but low flexibility.

We worked on support for the external change-over for the surface mounting equipment, which is considered the main obstacle to productivity improvement. We therefore studied a support system capable of optimum work arrangement based on

the simulation of actual production and real-time work rearrangement in response to changes in actual production conditions.

Introducing a support system for human-machine coexisting environment is one of the ideas to improve the resent situation in manufacturing.

2 Visualization of Change-over Activities

2.1 The Background

We manufacture various types of PWAs (PWB Assemblies) in order to fulfill the requirement from various industrial market fields. There are three PWB assembly lines operated in three shifts.

Basically, the PWB assembly line is fully automated, but still requires a lot of manual works especially for the external change-over. Major activities of the external change-over are settings of parts supplied by reeled packages, which include searching for the parts, fixing the parts to feeding mechanisms, and so on. Fig. 1 is an example of the external change-over activity.



Fig. 1. One of the external change-over activities: Fixing parts to feeding mechanisms

Such feeding mechanisms are highly dependent on the individual specifications of machines. Despite all of the machines in our site manufactured by the same company, the feeding mechanisms are different, and can not share them among machines. Because of the great amount of PWA types herein mentioned, we have to handle 10,000 or more items of parts in the daily operations. Doing monotonous operations under the presence of many similar parts, likely induces the luck of concentrations, or fatigues.

This is assumed to be the reason why:

- Surface mounting machines often cannot start operations because of waiting for the completion of the change-over.

- Many checking processes are required to avoid miss placements of parts.
- Some parts often become the targets for scrambling among operators.

There are almost 11 years of history for PWB assembly in our company, so people involving the production engineering, and perhaps the line leaders have their own ideas to improve the above situations. They have already adapted some ideas, but have not proven yet. A holistic approach to evaluate the effect of *kaizen* (= improvement) is required.

2.2 Steps for Evaluation and Implementation

Based on the above background, we are now on the half way of reform our PWB assembly lines.

The following is our general approach to reform the manufacturing lines. The approach is composed of several steps, such as:

- (1) as-is analysis by IDEF0,
- (2) to-be analysis also by IDEF0,
- (3) structuring a process flow by IDEF3, and
- (4) evaluating the above process flow by simulation.

These steps form the evaluation part in our approach. Furthermore, the following steps are also included, which form the implementation part, namely:

- (5) making a concept of a support system to realize the to-be process,
- (6) evaluating the system's prototype in a real manufacturing line,
- (7) implementing and running the system, and
- (8) improving continuously.

Hereinafter, we will describe (1) – (5) steps, with focusing on the external change-over, which is the most important issue affecting QCD (Quality, Delivery, and Cost) in PWA manufacturing.

3 Evaluations of the External Change-over

3.1 IDEF0 Analysis

Fig. 2 is the as-is model of the external change-over in our PWB assembly lines before the latest reform. This model is written in IDEF0 [1] modeling format. IDEF0 is one of the most famous methods for analyzing business functions, and supports to redefine individual business functions by visualizing activities and resources included in the functions.

This as-is model contains an obvious time-wasting activity, “Searching for the missing parts,” which has no control to regulate, and no mechanism to support operator's working processes.

The to-be model in Fig.3 is the consequence of elimination of the time-wasting activity from the as-is model. Here, we have introduced a change-over support system to indicate expected parts locations with the daily manufacturing schedule. Operators can check the actual parts locations, and statuses such as “being used in the other lines,” “being attached to another machine's cassette,” and so on. Visualizing parts locations is effective to reduce meaningless hiding games of parts among operators.

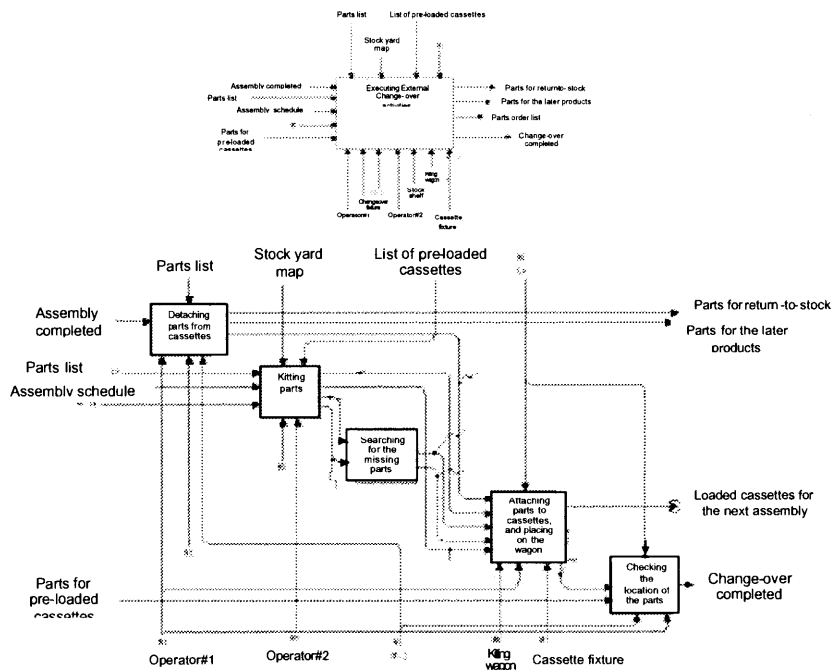


Fig. 2. The as-is model of external change-overs

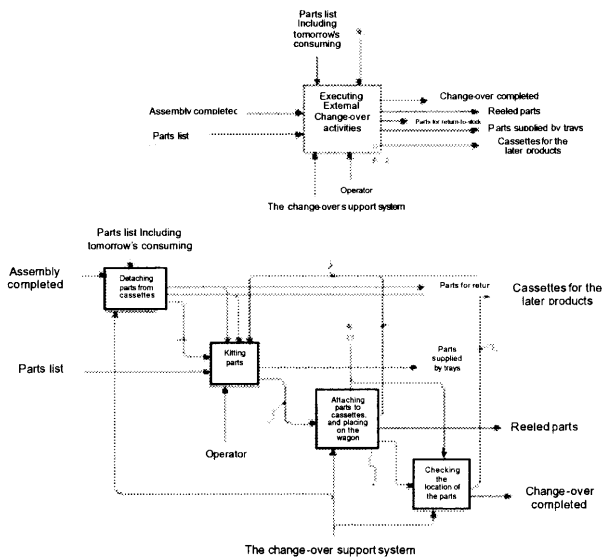


Fig. 3. The to-be model of the external change-over

3.2 Process Flow by IDEF3

IDEF0 analysis is effective to cut off obvious wastes or losses in business functions, the change-over support system already reduced at least 26 % of the total change-over time (in 9/2000) [2], and became the mandatory infrastructure for our PWA manufacturing. However, still our surface mounting equipments wait for the change-over completions for mostly an hour or more, so we need further more analysis beyond the methodological limitation of IDEF0. For instance, IDEF0 method does not include any chronological concepts and process orders, those are essential items when we analyze the details of manufacturing processes.

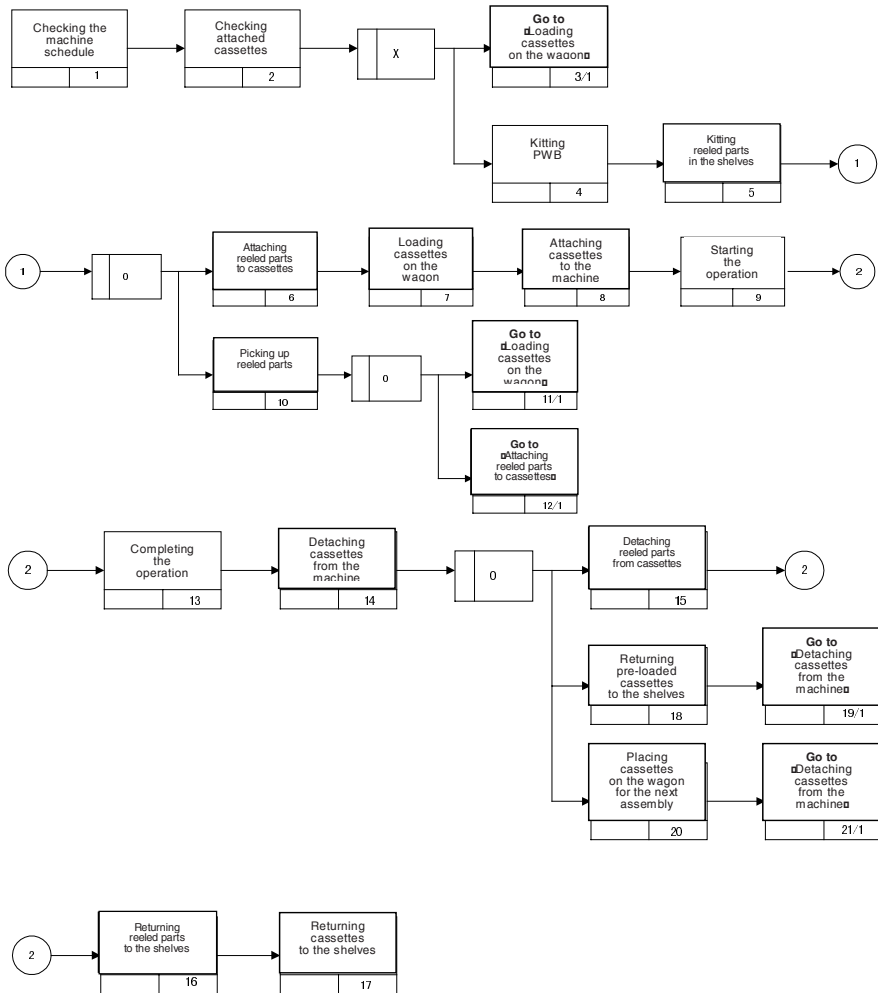


Fig. 4. The present process flow of the external change-over

Instead of the actual execution processes, idling processes are also included in the one-hour change-over, and occupy not neglectable amount in such volume. We have described the present process flow reflecting the results of the IDEF0 analysis (see Fig.4) by IDEF3 [3] for extracting the features of idling processes, and solving the problems. IDEF3 can complementarily use with IDEF0 for a process analysis in detail. We used IDEF3 to classify the types of processes, and also to get a draft of simulation model for evaluating individual processes.

From the process flow in IDEF3, we have discovered that the external change-over activities contains:

- (i) many processes whose order may be changed, and
- (ii) many processes that can be performed concurrently, instead of many ordinary processes.

We also have discovered that most processes can be performed by a single operator, but

- (iii) some processes require multiple operators.

3.3 Simulations and Analysis of the External Change-over

We reorganized the IDEF3 model in Fig.4, and constructed a simulation model. Fig.4 represents a complete process flow, but the simulation model consists of a group of individual process models, which we have named unit-processes, for allowing concurrent execution of processes in simulation. A unit-process is based on a general internal model of simulation used in machining process simulation. Fig.5 (a) shows such a general model, and also (b) is an example of a unit-process.

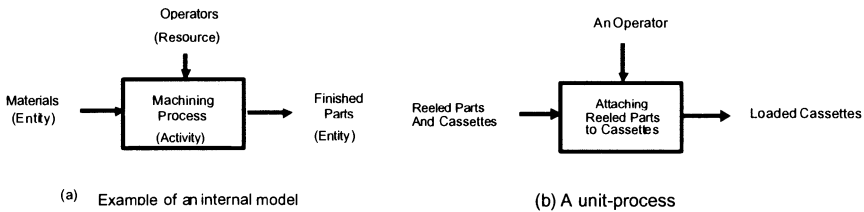


Fig. 5. The internal model: a unit-process

We have evaluated the unit-processes by Arena [4], one of the well-known simulation environments, but it was a trial version that can handle only limited numbers of models and pieces of data. Therefore, the results of applications of a full-fledged version to production sites need to be evaluated in the future. However, the results of applications this time gave us a rough idea. This is, we found that it is possible to estimate the daily change-over time by changing the order of processes mentioned in (i) and (ii) and to predict the time of entry into the processes mentioned in (iii), in which operators need the assistance of other operators.

Fig.6 is a screen shot of the simulation of the unit-processes.

Meanwhile, on the issue of change-over support from the viewpoint of reducing the

idling times, we have recognized the following limitations of the conventional simulation:

- (iv) The number of reeled parts used varies widely. Therefore, when operators are assigned to individual product models to be manufactured, it is difficult to even out work loads among the operators.
- (v) Actual work can be affected by external factors such as malfunctions of the machines. Therefore, even if the time when assistance is required is known in advance, operators cannot move on time.

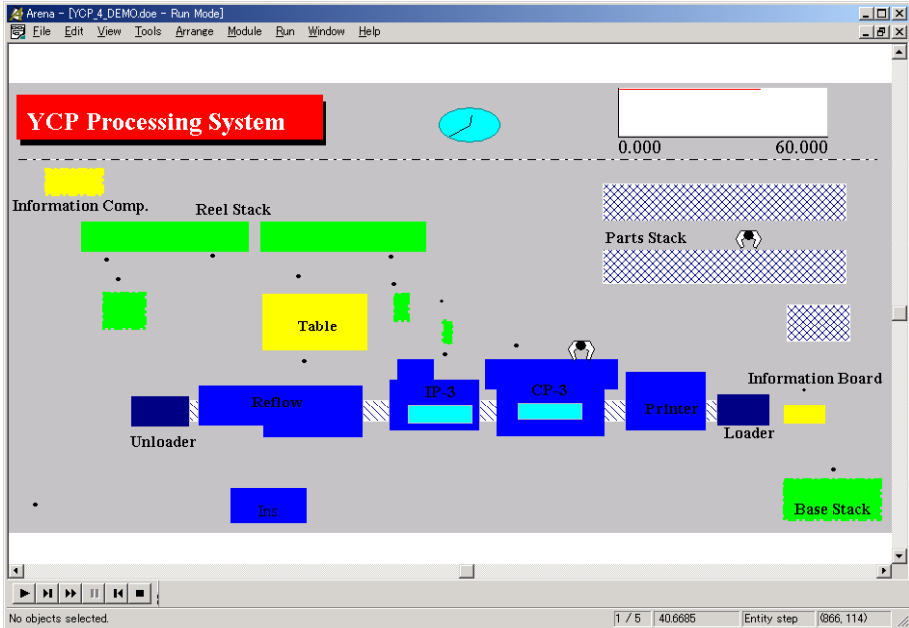


Fig. 6. A screen shot of the simulation

These problems can be solved if the field conditions can be visualized and grasped in real time. That is, if the field conditions are visualized, the problem mentioned in (iv) can be solved by assigning operators to individual processes in real time, and the problem mentioned in (v) can be solved by having operators available at that time carry out troubleshooting or extend assistance.

4 Introducing of a Real-Time Scheduling System

4.1 The System Overview

The production auction system [5] developed by Osaka University research group in IMS HUMACS project, is one of the choice to realize a real-time scheduling system.

The production auction system is originally designed for assign the individual machine times to operators through an auction-like mechanism.

With respect to a model required for linkage with the auction system, we considered that a unit-process can be reused as the core model with adding a new feature that the model completely independent from any specific performers, and named it the Unit-Process model (the UP model). Thanks to this independence, the UP model that has become executable shift into the execution state after capturing the resources (that is, performers) considered most suitable under the existing conditions. This makes it possible to utilize the flexibility of the auction system and thus allocate work and extend assistance flexibly.

Fig.7 shows the concept of real-time scheduling of the external change-over utilizing the production auction system.

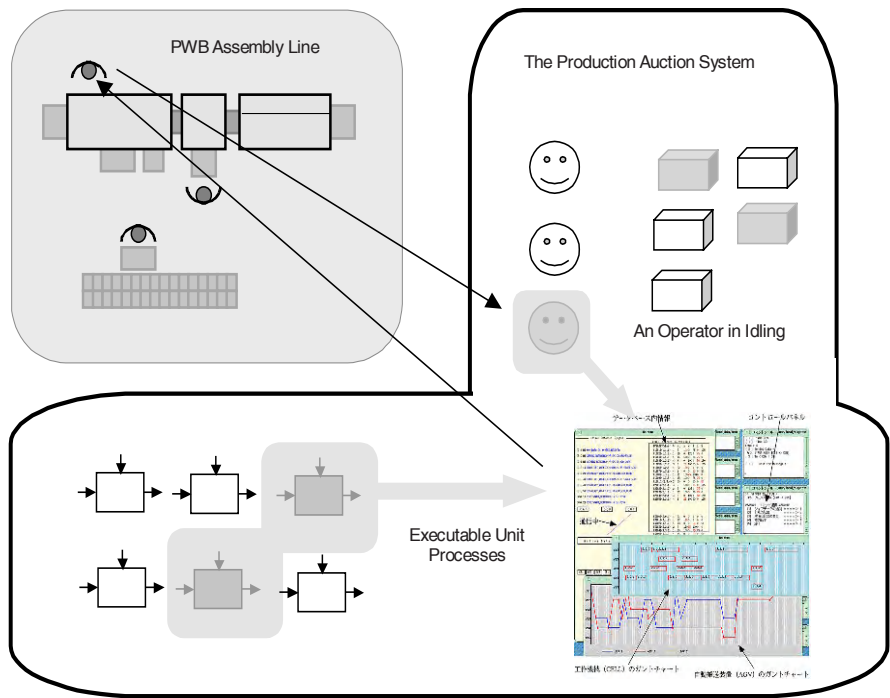


Fig. 7. Real-time Scheduling of the External Change-over

Real-time scheduling will enable operators to select somewhat independently what they should do from among candidates, without caring about whether the work is ordinary change-over or troubleshooting concerning parts. This will make it unnecessary for them to secure assistance by themselves. Furthermore, it could become easier for them to leave their place of work for a short time without caring about other operators around them. From the standpoint of manager, real-time scheduling will even out workloads among operators, and from the standpoint of operators, it will contribute to the realization of a better working environment.

4.2 Models for Real-Time Scheduling

Fig.8 describes the mechanism of actuating models within the real-time scheduling system.

The UP model has three different states, the sleeping state, the activating state, and the running state respectively. At the initializing phase of the system, all of the UP models are in the sleeping states. The UP model can be activated when receives the identifiers including the finished activities (i.e. the completed schedule-unit) from the other UP models representing the former process.

In the activating stage, the UP model receives a schedule-unit containing a list of parts numbers, the locations, and the status of the parts, and so on. Moreover, the UP model is waiting for the end of a bit, in which at least one or more operators or machines represented by the operator models or the machine models are assigned as action-owners.

All of the processes of the external change-over are done by human operators right now, but we imagine that introducing to change-over guiding mechanisms to ease operators work loads, or autonomous robots is one of the possibilities in the near future.

After receiving the above resources, the UP model will be in its running state. The UP model executes the schedule-unit received as one of the resources, and outputs the identifiers for the activation of the following processes.

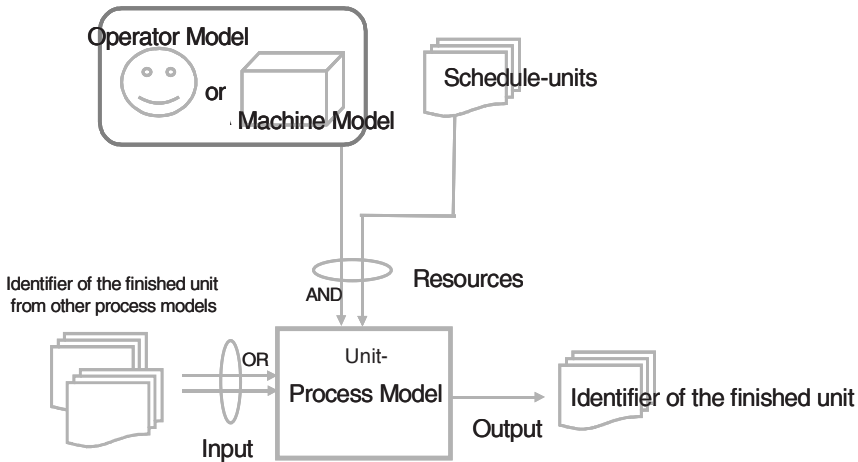


Fig. 8. Models for Real time Scheduling

Obviously, this kind of model execution mechanism can reduce idling of processes. However, on the other hand, to even out workload among operators and machines is one of our objectives mentioned in 4.1, this is somewhat incompatible with the basic idea of auction systems.

Some rules to choose action-owners from the list of candidates with avoiding competitions as far as possible, is one of our future issues.

5 Conclusions

In this paper, we have introduced our approach as an example to reform the real manufacturing lines in our site by using modeling and simulation technologies. We have focused on the external change-over, which is a collection of various different processes.

We divided the external change-over into the individual process units, and simulated them. We also studied a model structure required for supporting real-time scheduling based on the production auction system, and concluded that this system is effective as an alternative for meeting the requirement herein mentioned.

Further studies are required to realize the practical level of technical support that meets two seemingly contradictory requirements -- realizing an environment easier to work in and improving QCD -- at the same time.

This study has been conducted as a part of IMS HUMACS project.

References

1. D. A. Marca, C. I. McGown: IDEF0/SADT Business Process and Enterprise Modeling, Electric Solutions Corp. (1988)
2. IMS International Joint Study Program, WP1 deliverable of Participative Simulation Environment for Integral Manufacturing Enterprise Renewal (PSIM): As is Analysis at Yamatake Control Product (2000)
3. R. J. Mayer, T. P. Cullinane: IDEF3 Process Description Capture Method Report, Armstrong Laboratory (1992)
4. W. David Kelton, Randall P. Sadowski, Deborah A. Sadowski: Simulation with ARENA, McGraw-Hill Companies, Inc. (2001)
5. IMS International Joint Study Program, Results of Research on Organizational Aspects of Human-Machine Coexistence System (HUMACS) in Manufacturing Systems (IMS9005) (2001)

The E/S Tool

IT-Support for Ergonomic and Sociotechnical System Design

Martin Van De Bovenkamp¹, Ruben Jongkind¹, Gu Van Rhijn²,
Frans Van Eijnatten¹, Gudela Grote³, Jouni Lehtelä⁴, Timo Leskinen⁴,
Peter Vink², Scott Little³, and Toni Wäfler³

¹University of Technology (TUE), Faculty of Technology Management (TM), Eindhoven,
The Netherlands;

j.m.v.d.bovenkamp@tm.tue.nl
r.jongkind@tm.tue.nl
f.m.v.eijnatten@tm.tue.nl

²TNO Work and Employment, Hoofddorp, The Netherlands

p.vink@arbeid.tno.nl
g.vrhijn@arbeid.tno.nl

³Swiss Federal Institute of Technology (ETH), Institute of Work Psychology,
Zurich, Switzerland;

grote@ifap.bepi.ethz.ch
waeffler@ifap.bepi.ethz.ch
little@ifap.bepi.ethz.ch

⁴Finnish Institute of Occupational Health, Helsinki, Finland

timo.leskinen@occuphealth.fi
jouni.lehtela@occuphealth.fi

Abstract. A continuous changing environment imposes new requirements for assembly management and the role of human factors. To support an adequate response to these changes the European PSIM project was started in March 2000. This project aims at enabling the ongoing and integral improvement of assembly processes using advanced simulation software and up-to-date knowledge of sociotechnical systems design and ergonomics. The focus of this paper is on the development of the Ergonomic/Sociotechnical (E/S) software tool. Attention is paid to procedural aspects, the test of the first paper and pencil version and the development of the software prototype.

1. Introduction

Changing markets force businesses to produce increasingly more varieties of products. The customers become more demanding in terms of delivery time, reliability of delivery, quality and price. To cope with these high demands the assembly process management uses technical and organisational innovations to shorten order through-put time and reduce the number of mistakes. This leads to lower costs and customer demands can be better met. Therefore assembly process management is essential to competitiveness. Another crucial aspect is the availability

of ergonomically healthy workstations that support a motivated, efficient and healthy manner of working for employees. In the assembly industry an increasing awareness of this importance of human factors for the success of the company arises. A survey among 120 managers in the Dutch industry confirms this [10].

As an answer to the increasing importance of human factors in assembly enterprises, the European project PSIM¹ was launched in March 2000. PSIM stands for Participative Simulation environment for Integral Manufacturing enterprise renewal. The goal of the project, in which employees of industrial companies play a central role, is to enable the ongoing and integral improvement of company assembly processes using advanced simulation software and up-to-date knowledge of sociotechnical systems design and ergonomics. In one of the work packages of this project a software tool is developed that makes it possible for users to use state of the art sociotechnical and ergonomic knowledge for renewal of their daily work.

The ergonomic approach aims both at lead time reduction and at improvement of the human assembly tasks from an ergonomic point of view. Participation of company representatives in this approach is crucial, for reasons that have been discussed in previous papers on participatory ergonomics [6]. Another feature of the approach is that two disciplines are brought together: assembly engineering and industrial ergonomics. Previous projects demonstrate the surplus value of combining assembly engineering expertise with ergonomics expertise [5], [8].

The SocioTechnical System Design (STSD) approach considers social and technical factors, therefore making interactions apparent and allowing a joint optimisation that aims at avoiding technical biases in system design. Such biases not only neglect the potential of the human factor but – in the extreme – even destroy human potentials. Instead, a system design is aimed at, that explicitly considers the differences in strengths and weaknesses of both human and technical factors, in order to reach a new quality that could not be reached with the human or the technical factor alone. Besides this, a participatory approach allows the employees from different levels of hierarchy and different professional backgrounds (operators, supervisors, managers and engineers) to develop design solutions together. Consequently all experience and knowledge of the involved staff can be taken into consideration in the problem solving process.

The focus of this paper is on both ergonomics and sociotechnics in assembly industries. In the PSIM project a partly integrated software tool is developed in which new aspects of both approaches were used: the E/S tool (Ergonomic-Sociotechnical). The paper and pencil version of the tool was tested in two pilot sites. A software version of the tool will be tested at pilot sites as well. In this paper the procedure of this E/S tool is described as well as the test of a first paper and pencil version, its results and the IT demo version of the E/S tool.

¹ EU IST-IMS PSIM Project (IMS 1999-00004) funded by the European Commission and the Swiss Federation. PSIM is part of HUMACS, a project within the international IMS research program. Partners in the PSIM-Consortium are: TNO (NL) (project lead), Baan Development B.V. (NL), Data Consult S.R.L. (I), TUE Eindhoven University of Technology (NL), Institut für Arbeitswissenschaft RWTH Aachen (D), Chalmers University of Technology (S), University of Patras (Gr), Finnish Institute of Occupational Health (FIN), ETH Swiss Federal Institute of Technology (CH), C.R.F. Società Consortile per Azioni FIAT (I), Finland Post Ltd (FIN), Volvo Car Corporation (S).

2. The E/S Tool: The Combination of Ergonomics and Sociotechnics

The ergonomic and sociotechnical approaches are partly integrated in the E/S tool. However, the development of the content is quite specific for both approaches and was therefore mainly separately done by specialists. The developed E/S tool is a software tool that helps in the description/visualisation and evaluation of current and future assembly processes on ergonomic and sociotechnical aspects. It takes factors like sociotechnical analysis and design, physical load, process flow, mental load and ergonomic hazards (safety) into consideration. Figure 1 graphically outlines the E/S tool.

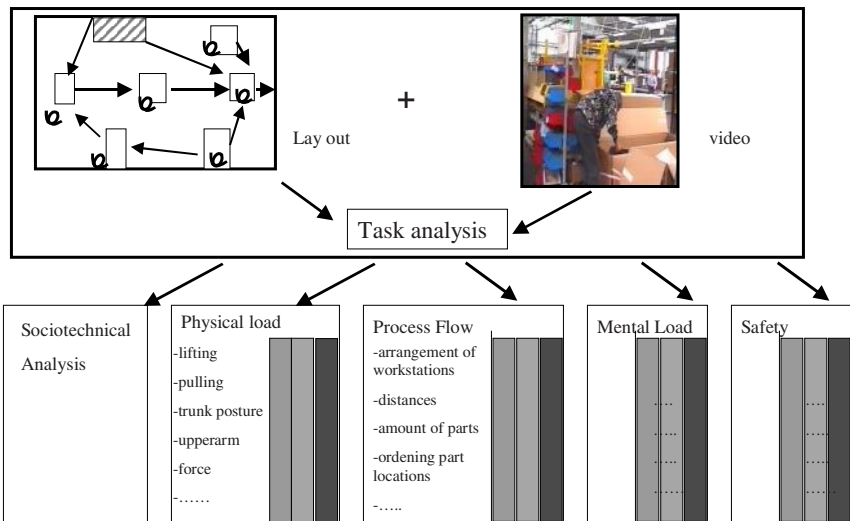


Fig. 1. Outline of the E/S tool: a common Task Analysis, followed by five different modules.

There is a common module in which a Task Analysis of the current situation is executed. This common module represents the integrated part of the tool. Why were these two different tools, with different scientific roots, integrated? When the ergonomic and sociotechnical experts met and compared their participative approaches and the way practical problems are tackled in both fields of study, they concluded that the first phase of both approaches is quite similar: analysing the current situation at a work system in terms of work division and lay out of the work stations. So it was decided to prevent from working twice by executing the Task Analysis for both approaches. Moreover this idea is in line with the central PSIM idea of developing one organisational model with shared databases that can be used by different (new or legacy) tools.

The shared Task Analysis is followed by five different modules: one sociotechnical module and four ergonomic modules. Based on the current state of knowledge the following four ergonomic modules have been made:

1. A process flow module.
2. A physical workload module.
3. Other health hazards module.
4. A mental workload module.

The first module evaluates process flows between different workstations and on every workstation according to guidelines (red/yellow/green). Aspects like arrangements of workstations, distances between workstations, amount of parts, ordering of part locations are considered.

The second module evaluates the physical load in every assembly station according to guidelines (red/yellow/green). Green means “safe”, yellow means “some risks, so measures must be taken” and red means “a lot of risks, so measures must be taken immediately”. Aspects like lifting loads, pushing and pulling, static working posture, repetitive movements and hand forces can be evaluated. In this evaluation module the most recently developed knowledge and standards on physical load are incorporated.

In the third module other hazards can be evaluated concerning safety, environment and noise using the checklist (red/yellow/green).

In the fourth module the mental load can be evaluated on the basis of a check-list (red/yellow/green) and a cubic model for mental load in which three dimensions play an important role: characteristics of activities (knowledge or routine based); time occupied and amount of task set switches.

The fifth module, the sociotechnical module, guides the user group step by step through a solution development process. For this process it is essential to analyse certain aspects in detail, to learn about system aspects, but also to consider the system as a whole, to make users understand the properties of their whole work system. Therefore the sociotechnical module supports a procedure that takes both an analytical and a holistic perspective into account by integrating two sociotechnical approaches. Whereas the Integral Organisational Renewal (IOR) approach (e.g. [1], [4], [9]) focuses on the holistic perspective, the KOMPASS method (e.g. [2], [3]) focuses on the analytical perspective. The distinction between the analytical and holistic perspective in these two approaches is not strict however; it is rather an accentuation of focus.

The sociotechnical module is flexible regarding the starting point of tool use. Starting points of tool use can be problems related to production processes and work organisation, the evaluation of already existing ideas for redesign or the general goal to optimise production processes and work organisation.

Analysing ergonomic aspects of work situations as well as interactions between social and technical factors and developing a solution that takes this into account is a complex undertaking. To support the tool users in this task the E/S tool is conceived as a software tool with specific features that help dealing with this complexity. By means of visualizations ergonomic aspects of work situations and relations between sociotechnical aspects are made transparent. This makes ergonomic and organisational redesign comprehensible for all employees.

3. The Procedure of the E/S Tool

The E/S tool consists of three main parts: the Task Analysis, the sociotechnical module and the four ERGO modules. The procedure of tool use consists of two main steps:

- 1) the use of the PSIM environment that leads tool users to the right tool and modules
- 2) the use of the modules themselves.

The first step is part of the PSIM procedure and is not further scrutinised in this paper. An essential aspect of this PSIM procedure is that the tool users are led to the right tool, means the tool that addresses their specific problem situation. The second part of the procedure, the use of the tools, is relevant in this paper. The procedure needed for use of the Task Analysis, the sociotechnical and the ERGO modules, consists of the following steps (figure 2).

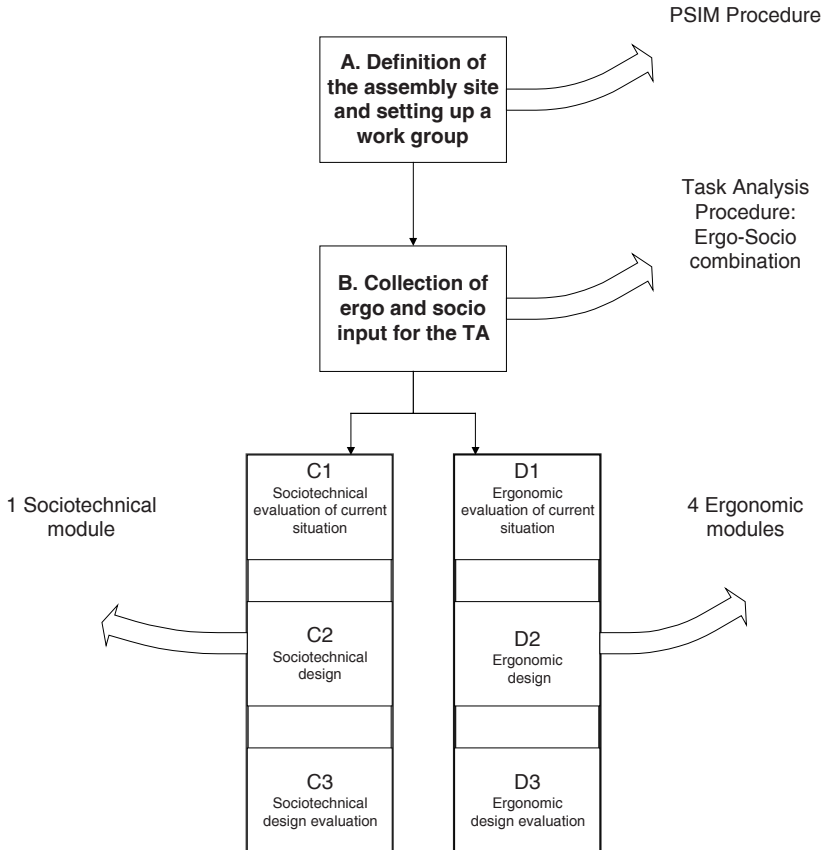


Fig. 2. Structure of the E/S tool

These steps are based on the approach Participatory Ergonomics [6] and on procedures derived from Integral Organisational Renewal (e.g. [1], [4], [9]). The E/S software tool supports these approaches in different steps (see figure 2):

A. Definition of the assembly site and setting up a workgroup.

This step is tackled in the PSIM Procedure as a general step to be taken by several tools.

- First, the assembly site that will be the object of the approach should be clearly defined. This assembly or test site might be (part of) a current assembly process or it might be (part of) a future assembly process, in which new products or new process steps are involved.
- Secondly, one should define the workgroup, that will be involved in the participatory process that follows. This group may comprise workers, manufacturing engineers, production managers, product designers, ergonomists etc.;

B. The collection of sociotechnical and ergonomic input for the Task Analysis.

This part constitutes the Task Analysis and is the integrated part of the E/S tool. Tool users perform the following activities:

- Making a graphical representation of the assembly lay out in the workgroup; This lay out has a hierarchy of three levels: lay out (1), workstations and transportation lines (2) and tasks performed on workstations (3).
- Making a video-record of each assembly process step and measuring data (if required for proper evaluation, the exertion of force should be measured as well).
- Making an ergonomic analysis (on the basis of the above activities). In this analysis an assessment of the current situation is made, on the basis of which the tool users fill in data that are stored in the E/S tool database.
- Making an analysis of organisational aspects. In this sociotechnical analysis tool users have to answer several questions regarding the current situation of the organisation. Their input is stored in the E/S database and will be used later on in the design phase of the tool. The tool presents visualized schemes of the unit of analysis in which the user group is asked to mark communication and process paths. Next to working on visualizations the user group has to answer questions concerning the unit of analysis. Employee related questions focus on the number of staff, the level of education, the tasks the employees have to perform, the number of employees directly involved in the production process, the number of employees not directly involved in the production process. Questions related to the organisation focus on the overall organisation (hierarchy levels, teams etc), communication paths, interfaces with other organisational units, wage system (are individuals or teams rewarded?). Questions related to the tasks focus on the task(s) of the unit of analysis, inputs and outputs of the unit of analysis considering information and material as well as the tasks of the individuals. Finally problems occurring in the unit of analysis have to be listed.

C. The sociotechnical module

The sociotechnical module contains three steps:

C1. Sociotechnical evaluation of current situation

In this step the user group defines the objectives they want to achieve and make an evaluation of the unit of analysis using the sociotechnical tool criteria provided by the sociotechnical module. The user group then relates the tool criteria to the objectives establishing the participatory simulation network. This phase supports the analytical perspective. The evaluation made in this step are automatically summarised in a report.

C2. Sociotechnical design

In this step the user group is provided with conceptual solutions as support for generating and designing concretised solutions for their problems. This step focuses more on the holistic perspective by offering conceptual solutions in terms of organisational structures that have certain characteristics. Tool users are stimulated to learn from these alternatives and to apply what's useful for their situation. Moreover the tool users are provided with the information they gathered in the Task Analysis to enable them to make an adequate design.

C3. Sociotechnical evaluation

In this step the design the tool users made is evaluated in terms of the tool criteria and the objectives that have been presented and formulated in step C1. A sociotechnical graphical network is used to support this evaluation. Finally a selection is made of a design that can be implemented.

D. The four ergonomic modules

The four ergonomic modules are used based upon the procedure outlined from D1 to D3:

D1. Evaluation of current situation on ergonomic aspects in the 4 ergonomic modules.

- Evaluation of physical load in every assembly station according to guidelines (red/yellow/green). Green means “safe”, yellow means “some risks, so measures must be taken” and red means “a lot of risk, so measures must be taken immediately”.
- Evaluation of process flow across different work-stations and on every workstation according to guidelines (red/yellow/green).
- Evaluation of mental load on the basis of a check-list (red/yellow/green) and a cubic model for mental load;
- Evaluation of safety and environmental aspects.

These evaluations are automatically summarized in a report.

D2. Generation of possible alternatives (improvements); This is done in the workgroup and supported by the E/S tool. Alternative designs are made and compared to each other.

D3. Evaluation of alternatives supported by the E/S tool and selection of a design that can be implemented.

Further on these procedures are clarified by presenting samples from the software implementation.

4. The Test of the Paper and Pencil Version of the E/S Tool

The paper and pencil version of the E/S tool consists of a written procedure of both separate tools and was first tested at two test sites of partners of the project before implementing it in software. The tests for the sociotechnical and the ergonomic modules were done separately. This because during the test, the procedures were still in development and changes in the contents of both tools were still possible. The tests were meant to find out if the procedures of the tools were workable for the tool users and if the content of the modules was not too complex for them. Both tests were performed in workshop sessions of three to four days. In general both the tool developers and the tool users were satisfied with the tests. For the tool developers the tests served as input for further development of the tools. For the tool users the tests were useful for a sociotechnical and/or ergonomic analysis of their current work situation and design of a new situation.

The sociotechnical part of the E/S tool has been tested at a production site of an industrial partner of the PSIM project. In the work shop of the test, eight employees from different levels of the hierarchy performing different functions participated. Three members of the sociotechnical team moderated the workshop.

The paper and pencil version consisted of three phases. For the first phase, an analysis of the current situation, a previously executed as is description was used. Therefore a problem analysis was not conducted during the workshop.

The as is description was read previously by the participants of the workshop and they agreed on the content of the formulated problems. The participating manager presented two ideas for solutions regarding the problems defined in the as is description (job rotation and self managed work teams). The participants were asked to derive objectives from the proposed solutions and the defined problems. The objectives then were clustered and prioritized. This first phase was considered by the participants as a typical way to approach problems and to discuss ideas of solutions.

In the second phase of the test, it was necessary to provide the tool users with information on the theoretical background of the STSD-approach (a short and simple introduction) and to explain in more detail the used tool criteria. The tool criteria were considered useful and the steps of this phase as a new way to look at the situation. Relating the objectives to the tool criteria at first was not clear to some participants and needed further explanation. In order to explore interdependencies of criteria and objectives the relations defined by the participants were entered into a mock-up version of a part of the tool.

During the third phase the two ideas for solutions suggested by the management were elaborated and evaluated by means of the relations between the criteria and the objectives as modeled in the mock-up. The solution 'self managed work teams' turned out to have a stronger positive impact on the objectives than the solution 'job rotation'. Due to time constraints it was not possible to develop new solutions from scratch but as the suggested solutions were merely basic ideas there were enough degrees of freedom to implement own ideas. The information presented for supporting this phase was regarded very helpful and easy to understand.

A paper and pencil version of the ergo tool has been tested at a company of VOLVO (test site 1) and at Finland Post (test site 2). The results of the test in two test sites show that the tool is useful. Possibilities for these companies to improve were

established for process flow, physical workload and other health hazards (see table 1). The test also showed that the mental workload module needs to be simplified for participative use with the work force.

Table 1. Possibilities for improvement on process flow, physical workload and other hazards in 2 test sites. In the process flow 12 items are evaluated green, yellow and red. Test site 2 shows more possibilities for improvement than test site 1. In physical workload 16 items are evaluated, which were mostly yellow or green. However, in test site 1 shoulder/back posture could result in complaints and in test site 2 lifting could cause problems

Test site		green	yellow	red
1	process flow	10 out of 12	1 out of 12	1 out of 12
2	process flow	5 out of 11	4 out of 11	2 out of 11
1	physical workload - shoulder/back	12 out of 16 items	1 out of 16 items	3 out of 16 items
	- shoulder	3 out of 8		5 out of 8 zones
	- back	4 out of 8		4 out of 8 zones
2	physical workload (lifting)	11 out of 15 items	2 out of 15 items	2 out of 15 items
2	other hazards (falling objects)	13 out of 23 items	8 out of 23 items	2 out of 23 items

5. The IT Demo Version of the E/S Tool

After having tested the first paper and pencil version of the E/S tool, it was programmed in a software demo version. This programming work was split up in three parts: the development of the sociotechnical module, the development of the ergonomic modules and the development of the Task Analysis (for both the sociotechnical and the ergonomic modules). The final E/S demo tool however is one tool in which the three parts are put together: an integrated Task Analysis makes it possible to go to either the sociotechnical or the ergonomic modules. The developed software tool makes it possible to work in a participatory way on ergonomic and sociotechnical renewal with few support from specialists. The E/S tool fits in a large PSIM environment in which more tools can be added. The PSIM environment takes care of all steps necessary to guide the tool user to the right tool for the problem or need he/she has. After the right tool has been determined the tool user is guided to the Task Analysis (TA). In this TA data of the current situation has to be filled in the tool and is stored in a database. This data is used further on in the tool. After going through the TA the ERGO- or sociotechnical modules can be entered.

Starting point of the evaluation is a lay out of the assembly line (figure 3). The lay out contains workstations and lines of transportation.

If you click on a workstation or transportation line it is possible to describe tasks performed at that workstation or transportation line (figure 4).

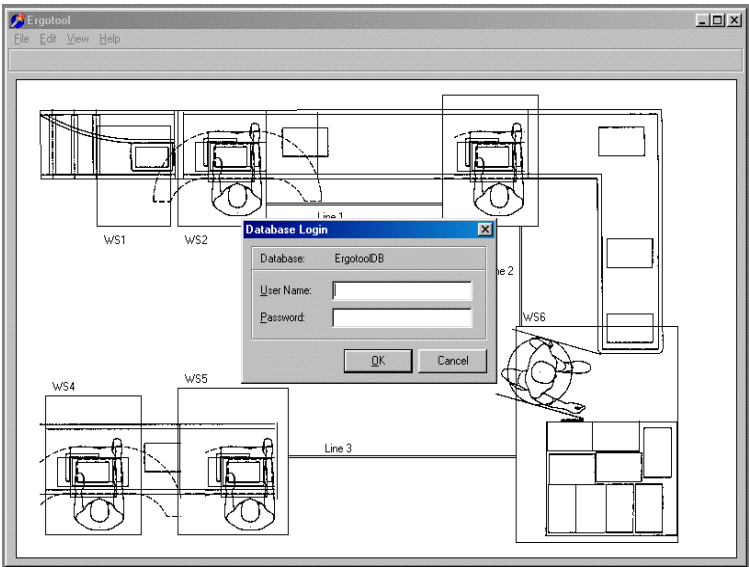


Fig. 3. Starting point is a lay out with workstations and transportation lines

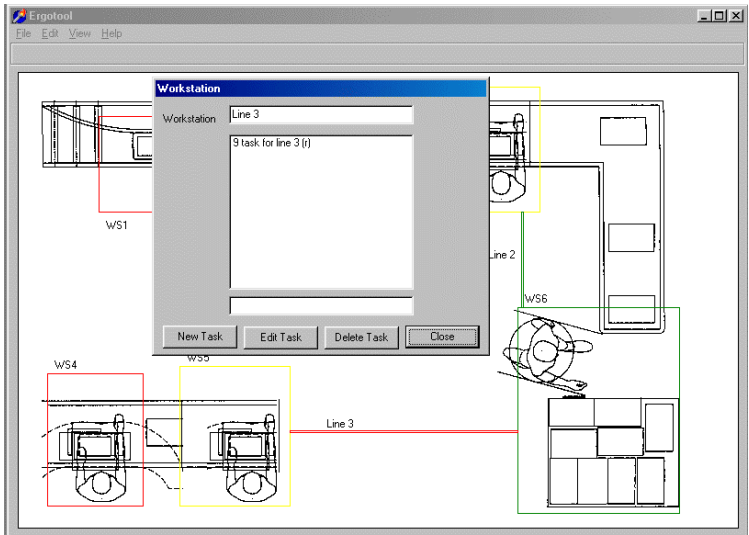


Fig. 4. Description of tasks on every workstation and line

6. The ERGO Modules

After a brief task description it is possible to show a video of the tasks. Then these tasks can be evaluated on relevant aspects: process flow (figure 5), safety and

environmental factors (figure 6), physical load (figure 7) or mental load (figure 8). After this evaluation workstations and transport lines in this lay out will be coloured green, yellow or red depending on the evaluation (figure 4).

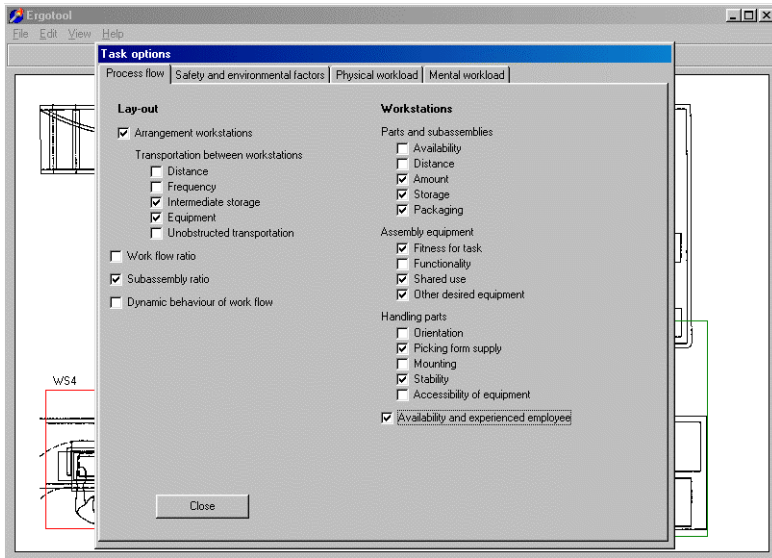


Fig. 5. Evaluation on process flow.

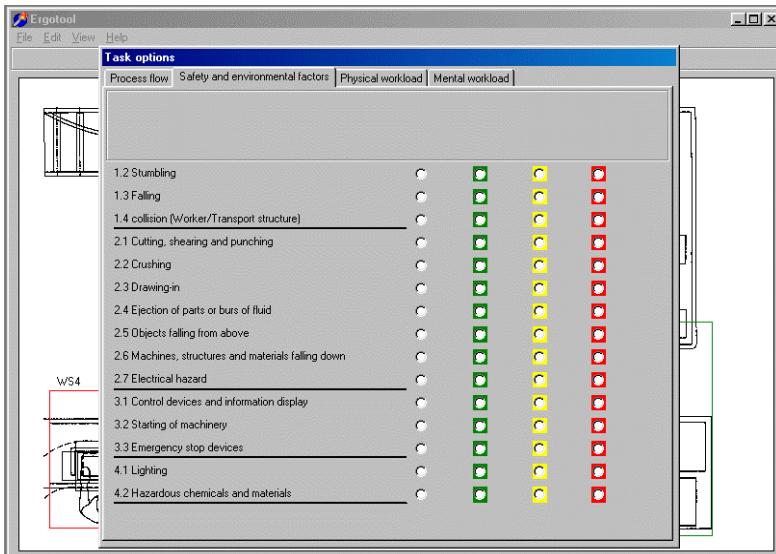


Fig. 6. Evaluation on safety and environmental factors on the basis of a checklist filled in by a group. “Under” every aspect there is a short explanation.

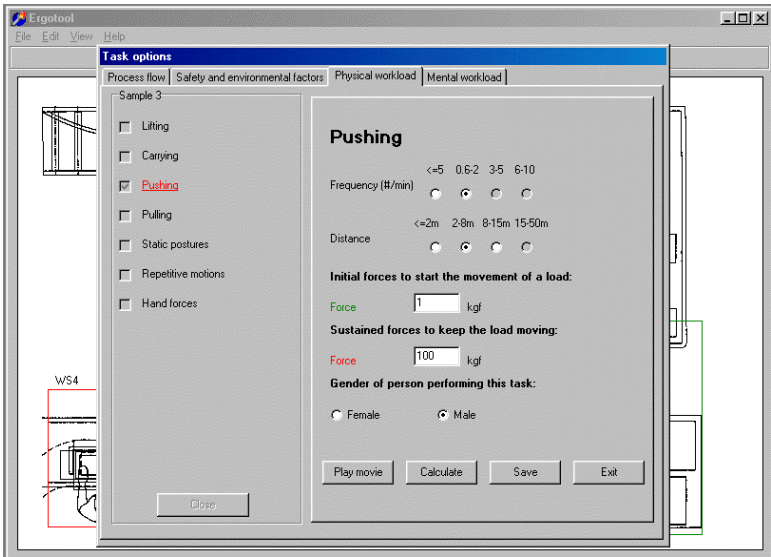


Fig. 7. Evaluation of physical load. An example is pushing (of carriers). Pushing can be evaluated by putting data like forces and distances. The system gives an answer in green, yellow or red.

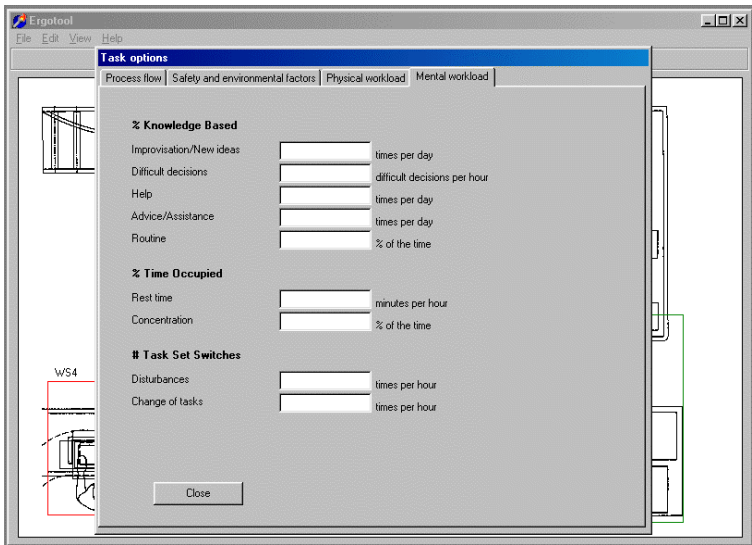


Fig. 8. Evaluation on mental load. Data can be entered in a group session. The system gives an answer in green, yellow or red.

7. The Sociotechnical Module

In the Task Analysis the sociotechnical module can be started. The software module consists of a number of steps:

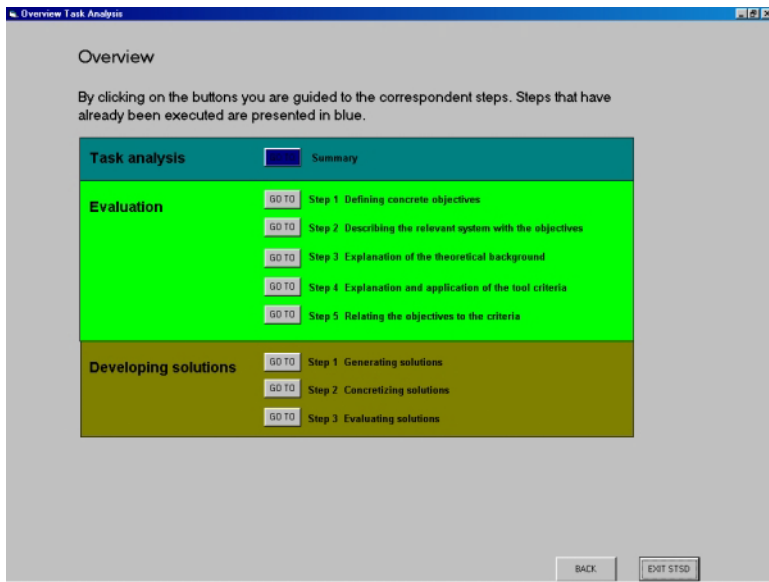


Fig. 9. Overview of the steps of the sociotechnical module.

By going through these steps the tool users work on generating ideas for a sociotechnical renewal of their current work organisation described and briefly analysed in the Task Analysis. During this process different aspects of sociotechnical organisational renewal are presented and used in the design. By offering this knowledge in an interesting way (with the help of pictures and animations) the learning process is stimulated with which the tool users easily increase their knowledge. With this knowledge, the tool users work on generating solutions for the problems they have and create a first concept of a redesign. This is done by answering questions and filling in these answers in the tool. Hence, by offering a sociotechnical framework that is easily understood by the tool users they can make their tacit knowledge explicit.

8. Conclusions and Next Steps

The E/S tool is unique regarding the combination of sociotechnical and ergonomic knowledge that is offered to the tool users in a software environment. In this way the state of the art knowledge of both fields is combined and offers possibilities for interactive learning in a modern ICT environment. Besides this, it makes a

participative use of the tool possible in which a group of employees from different hierarchical levels work on improvement of the current work situation.

The first paper and pencil tool test was considered as successful by both tool users and tool developers. In two industries the ergonomic part of the E/S tool proved to be successful in evaluating and identifying problems in the field of physical load, process flow, mental load and safety. The test of the tool demonstrated the value of involving work staff in companies and working in these groups on organisational problem solving, supported by an ICT tool. This enabled the users to find solutions which were accepted by all stakeholders. On the basis of these test results the final tool characteristics have been defined and a demo of the E/S tool was programmed in software. This tool can be used by workgroups as well as experts (both sociotechnical and ergonomic).

The first software prototype will be tested within the next months and will provide the tool developers with valuable information that can be used in the further development of the E/S tool. Currently the E/S tool is further integrated with the PSIM environment to create one total PSIM tool. This integrated tool will also be tested at one or more test sites before the end of the PSIM project (March 2002) and will be further developed to a demo version.

References

1. Eijnatten, F.M. van, & Zwaan, A.H. van der (1998). The Dutch IOR approach to organisational design. An alternative to Business process Re-engineering? *Human Relations*, 51 (3) March, 289-318.
2. Grote G., C. Ryser, T. Wäfler, A. Windischer, S. Weik, & M. Zölch (1999), *Wie sich Mensch und Technik sinnvoll ergänzen, Die Analyse automatisierter Produktionssysteme mit Kompass*, vdf Hochschulverlag AG an der ETH Zürich.
3. Grote, G. (2000), PSIM: Participative Simulation environment for Integral Manufacturing enterprise renewal, Invited sessions for 8th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Human-Machine Systems (HMS), Swiss Federal Institute of Technology.
4. Kuipers, H., P. van Amelsvoort (1995), *Slagvaardig Organiseren*, Kluwer Bedrijfswetenschappen, Deventer.
5. Looze, M.P. de, G. van Rhijn, G. Tuinzaad and J. van Deursen (2000). A participatory and integrative approach to increase productivity and comfort in assembly. *Proceedings of the XIVth Triennial Con-gress of the International Ergonomics Association (on CD-ROM)*, 5-142, San Diego.
6. Noro, K. and A. Imada (1992). *Participatory Ergonomics*. Taylor and Francis, London.
7. Rhijn, G. van, M.P. de Looze and B. Tuinzaad (2000). In: *Design of efficient assembly flow and human centred workplaces in Dutch assembly com-panies*. *Proceedings of the 5th Int. Workshop on simulation games in production management* Zulch G. and A. Rinn (Eds), 163-172. Karlsruhe.
8. Sitter, L.U. de (1998), *Synergetisch produceren*, Van Gorkum, Assen.
9. Vink, P. and M. Peeters (1998). Balancing organisational, technological and human factors. In: *Human Factors in Organisational Design and Management-VI*. P. Vink, E.A.P. Koningsveld and S. Dhondt (Eds): Elsevier, Amsterdam.

Construction of Virtual Working Environment and Evaluation of the Workers

Kageyu Noro and Ryohei Tanaka

Graduate School of Human Sciences Waseda University,
2-579-15 Mikajima Tokorozawashi Saitama, Japan
Kageyu Noro: HAG02334@nifty.ne.jp

Abstract. In the age of Information Technology, the work styles of office workers are transformed. In this study, working environments for new work styles were constructed virtually. Then a method for the evaluation of the worker in that environment was developed. Workers were classified with Interaction and Autonomy, and workers with “High Interaction and High Autonomy” and “Low Interaction and High Autonomy” were evaluated. As a result of the evaluation, the following are suggested for the work environment design. 1. An office layout which makes workers concentrate on their own work, and also allows the worker to choose when to rest with a reclining chair. 2. For the “High Interaction” worker, a layout that helps worker’s communication with others is preferred.

1 Introduction – The Transition of Work Style –

The advance of information technology changes both living and work environments. The effect is especially remarkable in the work environment. It can be expected that technology will continue to change our concept of work in the future.

In the 1900’s, simple repetitive work, which ignored worker’s autonomy, became the norm for workers for a long time. In the 1950~1970’s, standardization and efficiency of work was an objective and, as a result, work was constrained by rules and manuals. Then in the 1980’s it was recognized that human behavior is based on one’s autonomy. In the late 1990’s, work depending on worker’s decision-making and autonomy increased. Saito (1998) described the above-mentioned transition of work style in the form of a figure (Figure 1).

Though work which is performed facing a computer monitor still takes place in office, a new work style has appeared in “The Age of Information Technology”. This work takes place not only inside the office but in meetings, through collaboration at a plant site, or in the outside world to collect information. In such situations, it is thought that the former studies and experiments to target VDT work are no longer representative.

The measurement system to evaluate new work styles and the way to reflect those results in workplace design must be clarified. The final purpose of this study is the development of an evaluation method that covers the above. For this purpose, fundamental and preparatory experiments are performed. The purpose of this paper is to

give an account of the construction of a virtual working environment, and a new evaluation system for workers who work in such environments.



Fig. 1. The transition of work styles (Saito, 1998). The first step in the figure is the work style in the 1900’s, the second step is 1950-1970, and the third step is the 1980’s and later.

2 Theoretical Basis

To classify the workers, an “Autonomy” and “Interaction” matrix is used in this study (Fig 2). This matrix is proposed by Steelcase Inc. Each axis of the matrix is explained below.

Autonomy: Depending upon how much the subject is free to decide where or when to work.

Interaction: In order to execute work, interchanges with others take place frequently

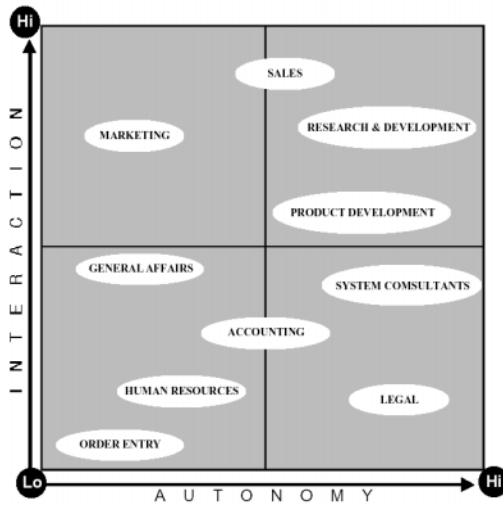


Fig. 2. Autonomy/Interaction Matrix (Steelcase General Catalogue 1997). This figure shows a distribution of job types classified with Autonomy and Interaction.

Normally, a classification of work styles must be based on objective research that needs enormous amount of research data. For that reason, figure 2, which is subjective base, was applied in this study. “High Interaction and High Autonomy” and “Low Interaction and High Autonomy” work styles are highlighted in the era of Information Technology. In order to discuss the characteristics and demands of the work place, it

is necessary to deal with each work style that is classified in the Matrix. Furthermore, the worker with “High Interaction” acquires information at an irregular rate. For that reason, measurements of workers’ responses are called for.

3 Method

Because scientific reports of “High Interaction and High Autonomy” work are scarce, it is necessary to consider a method of investigation. In this study, “Task Analysis” was chosen as the descriptive method to clarify worker’s work and behavior, and “Psycho-Physiological Evaluation” was conducted during the work and behavior.

3.1 Task Analysis

It is necessary to know the contents of the “High Autonomy” work that is found in the Information Technology age. Task Analysis is a description method for understanding the contents of work, behavior of worker, and operation of machines simultaneously. This method is often used for the analysis of accidents. In the work environment, the need for such an analysis method as declined in the 1980’s. To understand the “High Autonomy” worker, observation and analysis of behavior are necessary. Furthermore in Educational Engineering, the method is used in the investigation of autonomy of students. In this study, task analysis is used to describe the contents of work, and to clarify the characteristics of work styles.

3.2 Psycho-Physiological Evaluation

Worker s’ stress, fatigue, and mentality must be investigated. There are many studies reporting psycho-physiological evaluations. It goes without saying that many psycho-physiological studies of VDT work have been performed in the past. Psycho-physiological experiments tended to take place in restricted and simplified experimental environments to understand the relationship between “stimulus and response”. The reaction time experiment consisting of pressing a button after seeing a light is one such example. The experimental environment, is very different to the office environment where “High Autonomy” work takes place, because movements and tasks of subjects are restricted. Measurement data from the environment of actual offices are difficult to obtain with high reliability. For that reason, psycho-physiological measurements are compared with “Task Analysis” to consider the meaning of the results.

4 Experiment

To begin with, the future office was constructed virtually in order not to restrict the worker’s movements. For the “Task Analysis” of the worker, psycho-physiological responses are measured simultaneously with the observation of work and behavior.

The “High Interaction and High Autonomy” worker and the “Low Interaction and High Autonomy” worker are measured on this occasion.

5 Method of Experiment

5.1 Environment of Experiment – Construction of Virtual Working Environment –

The construction of a virtual working environment makes it possible to have an experiment for a worker with a non-constant task frequency. The layout of the virtual working environment is shown in figure 3. The conventional office, represented by an island type, is adapted to “Low Autonomy” workers, not to “High Autonomy” workers. For that reason, a virtual working environment that allows workers to choose their own work place and work contents was developed.

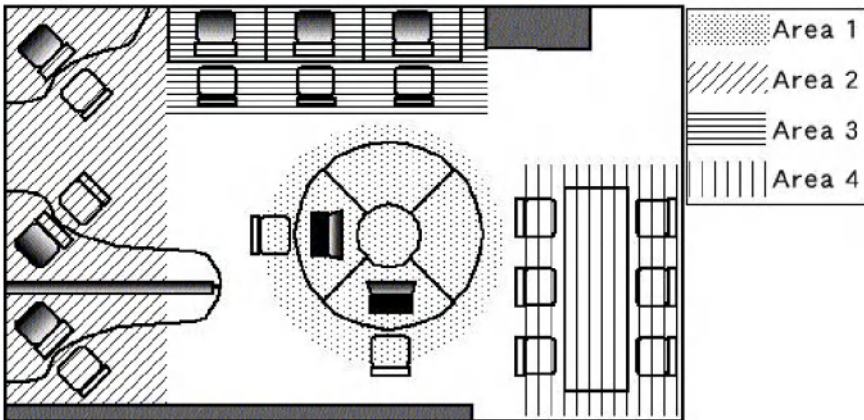


Fig. 3. The layout of virtual workplace is shown. Area 1 is constructed for a work place of a high interaction and high mobility worker, area 2 is constructed for a worker who needs to concentrate, area 3 is constructed as conventional work place, and area 4 is constructed for a meeting.

5.2 Psycho-Physiological Evaluation

The measurement time was 50 minutes, which is the recommended time for continuous working given in VDT Guidelines. The measurement apparatus was made to be wearable so as to restrict the movement of subjects as little as possible. The room was also video taped to observe movements and behavior of the subjects. The following are the measurement systems used.

1. *Measurement of Pelvic Angle:* A system measuring inclination of the pelvis was developed. A sensor set contains a 3-dimensional gyro element and an acceleration sensor on the side of subject's waist.

2. *Measurement of Eye Movement:* To understand eye movements of a subject, EMR (Eye Mark Recorder) 8 (NAC) was placed on the subject's head. The EMR 8 does not interfere with the subject's natural movements and it gives less fatigue to the subject (Figure 4).



Fig. 4. Picture of Eye Mark Recorder 8 (EMR8). EMR records eye movement of subject. The recorder is lighter and does not constrain the subject's movement.

3. *Heart Rate (HR):* An HR monitor (NIHON KOHDEN) was used to measure heart rate. The wireless monitor does not restrict the movements of subject. Heart rate is expressed as beats per minute to measure long term variations.

4. *Expression of Subject's Face:* The expression of subject's face was video taped with a CCD camera (SONY).

5. *Working Posture:* To capture transitions of working posture, the subject's working posture was video taped from the side of subject.

6 Results

6.1 "High Interaction and High Autonomy" Worker

The following four types of behavior were seen in the "High Interaction and High Autonomy" worker, during the 55 minutes of measurement.

1. PC work
2. Conversation (Discussion)
3. PC work and Conversation (Discussion) are done at the same time
4. Rest

The contents of conversation were mainly conversation with staff. To capture the results of psycho-physiological measurements and task analysis, a storyboard was made (figure 5). The following are a summary of the results of measurements and behavior of the worker.

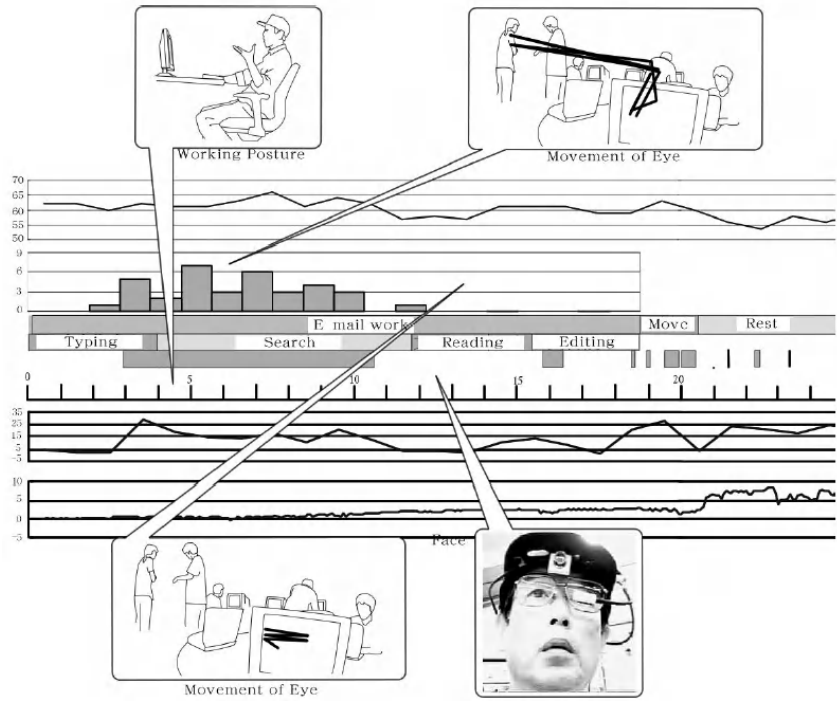


Fig. 5. Example of a Story Board. The center part of the figure is a flow of working contents. Using this figure, psycho-physiological measurements are evaluated together with the work.

Pelvic Angle: Little movement was seen during PC work (fig. 6). From previous studies of VDT work, body movements tend to be fixed during VDT work. In contrast to this, frequent movements of pelvis were seen during conversation and rest. The pelvis also tilted backward during rest. From the video of working posture, it was also found that the majority of movements and reclined backward postures were seen during conversation and rest.

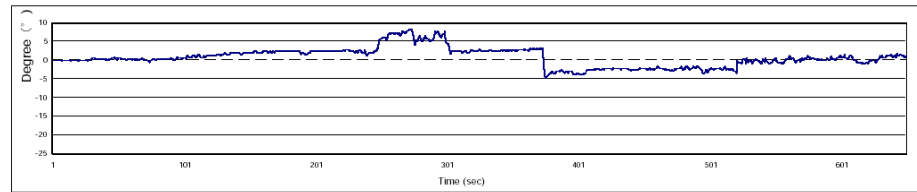


Fig. 6. Transition of pelvic angle (High Interaction and High Autonomy)

Eye Movements: During the behavior of PC work + Conversation, the subject's gaze comes and goes between the person being spoken to and the PC monitor. Even if the person being spoken to was not looking at the subject, the subject looked the back of the person while talking.

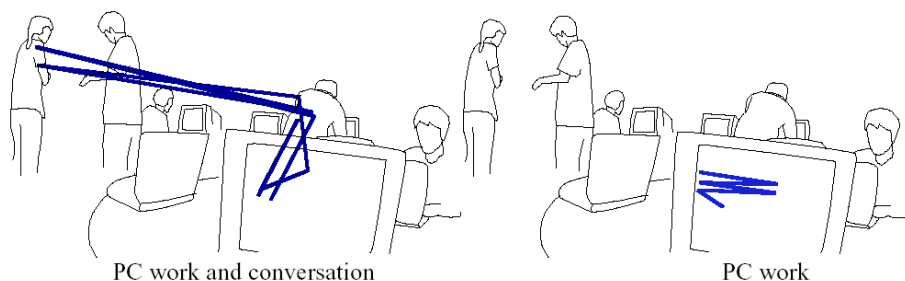


Fig. 7. Locus of Eye Movement (High Interaction and High Autonomy), difference between PC work + conversation and simple PC work.

Heart Rate: Figure 8 is the transition of heart rate during the 55 minutes. The lowest point was recorded during rest time and the highest point was recorded when PC work and Conversation took place at the same time.

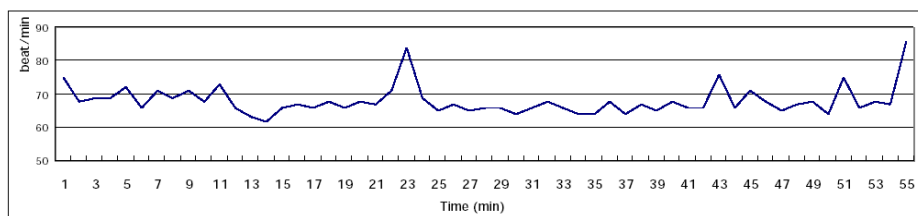


Fig. 8. Transition of heart rate (High Interaction and High Autonomy)

6.2 “Low Interaction and High Autonomy” Worker

The following two types of behavior were seen in the “Low Interaction and High Autonomy” worker, during the 55 minutes of measurement.

1. PC work
2. PC work and Conversation (Chatting) are done at the same time

In this case, the contents of conversation were not related to subject’s work. It was only chatting.

Pelvic Angle: Compared with the “High Interaction and High Autonomy” worker, the “Low Interaction and High Autonomy” worker’s range of movement was greater (fig. 9). From the video of working posture, the large movements on 14 minutes and 23 minutes were found to correspond to changing posture and reseating. Each reseat tilted the posture forward. The posture during PC work was mostly leaning forward, because subject worked with a track-point in the middle of a laptop PC keyboard most of the time.

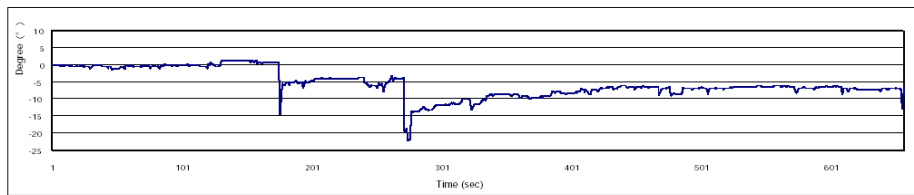


Fig. 9. Transition of pelvic angle (High Interaction and High Autonomy)

Working Posture: The video recording revealed not only the reseats corresponding to the pelvic angle data, but also there were many small movements of legs and arms. *Eye Movements:* During PC work, the subject mostly watched the PC monitor. During the behavior of PC work + Conversation, the subject’s gaze was found to come and go between the person being spoken to and the PC monitor.

7 Discussion

7.1 “Low Interaction and Low Autonomy” Worker

Figure 10 represents data collected by the Association of Employment for Senior Citizens (Takada, 1983). These are data from a “Low Interaction and Low Autonomy” VDT worker. This data are used for comparison data with “High Autonomy” workers.

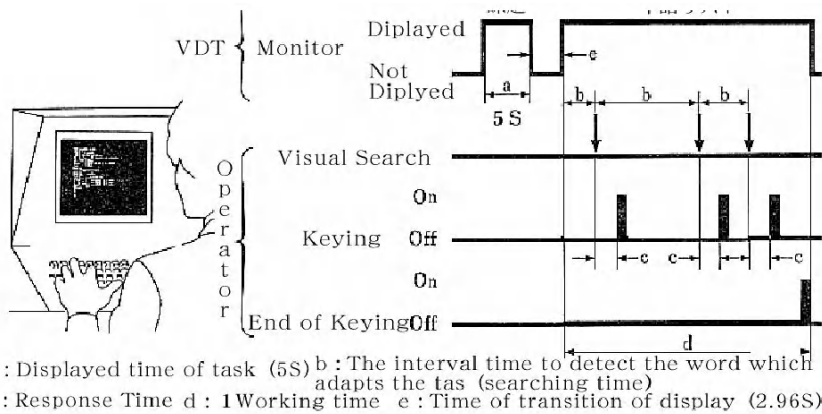


Fig. 10. Task Analysis of “Low Interaction and Low Autonomy” worker (1983) (Association of Employment for Senior Citizens)

The following are the main types of work at that time.

1. Watching Monitor
2. Searching a series of letters from a monitor (visual search)

The processing time of the computer was about 3 second longer than now. Moreover, a mouse was not used at that time. The characteristics of the “visual search” in the figure 10 were watching on monitor with concentration. The working posture of the worker was an upright posture. The PC work of the “Low Interaction and Low Autonomy” was given and constrained by rules and manuals. For that reason, most of the jobs were simple PC work. The reclining posture and conversations, as in “High Interaction and High Autonomy” work, were not be seen in work at that time.

7.2 “High Interaction and High Autonomy” Worker

In comparison with other work styles, the “High Interaction and High Autonomy” worker must deal with many types of work. For that reason, PC work and conversation must be done at the same time. A work place is needed where the workers can deal with different types of work simultaneously whilst concentrating on a primary activity. A workplace enabling the worker to take a rest easily is suggested. For example, a chair which can take on a reclining posture. Moreover, the workplace must easily enable communication with others.

7.3 “Low Interaction and High Autonomy” Worker

It was found that the “Low Interaction and High Autonomy” worker concentrated on his own work. Because the contents of conversation were chatting when PC work and conversation were done at same time, it was considered that such conversation was not for work but for refreshment. Unlike the “Low Autonomy” worker, thinking time while watching monitor was observed. Therefore, it is suggested that the “Low Interaction and High Autonomy” worker needs work place where he can concentrate on his own work.

8 Conclusion – A Suggestion for Design of Work Place –

The target of this study is “High Interaction and High Autonomy” work that is expected to increase in the future. For that reason, a future working environment was constructed virtually, and psycho-physiological measurements were made. As a result, each work style needs a different design of work place. The following are suggestions for workplace design.

- An office layout which makes the worker concentrate on his own work, and also allows the worker to choose when to rest in a reclining chair.
- For the “High Interaction” worker, a layout that facilitates the worker’s communication with others is preferred.

9 Further Research

The “Autonomy” and “Interaction” axes in figure 2 are subjective base. In this study, the matrix was applied to classify the work styles. From the results of this study, it was suggested that the axes could be measured by objective.

References

1. Murako Saito : Ergonomics on Organization and management. Workers behavior as an active perceivers in job environment. From passive behavior to adaptive action, Japanese Journal of Ergonomics, vol.34, no.6, 287-295 (1998)
2. Steven L. Sauter, and Lawrence M. Schleifer, Shiri J.Knutson : Work Posture, Workstation Design, and Musculoskeletal Discomfort in a VDT Data Entry Task, Human Factors, 33(2), 151-167 (1991)
3. Yukiyo Kuriyagawa, Ichiro Kageyama : A modeling of the Heart Rate Variability to Estimate mental Work Load at mechanical Operations, Japanese Journal of Mechanical Engineering, Vol. 66, No. 643 (2000), 836-842
4. Tutomu Takada : The Study of Work Adaptation for Senior Citizens, Association of Employment for Senior Citizens, 3rd edition (1983)
5. Aki Nakanishi : The Human Side of Knowledge Management, Office Automations, 40th Annual Meeting (1999)
6. Norihiro Yamakawa : Change of Human Behavior in Information Society, Office Automations, vol.20, No.1 (1999)
7. Japan Facility Management Association : Information Technology and Office, JFMA Current No.44 (2000)
8. Brunnberg H : Evaluation of Flexible Offices, Proceedings of the IEA 2000/HFES 2000 Congress, 1-667 (2000)
9. Nico Delleman, Mark Boocock, Bronislaw Kapitaniak, Peter Schaefer, Karheinz Schaub : ISO/FDIS 11226: Evaluation of Static Working Postures, Proceedings of the IEA 2000/HFES 2000 Congress, 6-442 (2000)
10. Shin Saito, Masaru Miyao, Takaaki Kondo, Hisataka Sakakibara, Hideaki Toyoshima, Ergonomic Evaluation of Working Posture of VDT Operation Using Personal Computer with Flat Panel Display, Industrial Health, 35, 264-270 (1997)
11. Chi-Shiun Wu, Shin Takeda, Hiroyuki Miyamoto, Kageyu Noro : Relationship between Seat Hight and Fatigue in Seated Posture during VDT works, Japanese J of Ergonomics Suppl. Vol.32, 342-343 (1995)
12. Kaoru Suzuki : Changes of Physiological Measures during VDT Work, Japanese J of Ergonomics, Supple Vol. 30, 306-307
13. Akio Koyama, Kazuo Saito : A Agenda for Measurement and Estimation of Physical and Mental Strain, Japanese J of Ergonomics, Vol.29, No.6 (1993)
14. Kageyu Noro, Ryohei Tanaka: Human Factors and Information Technology, Occupational Safety & Health, Vol.2, No.1 (2001)
15. Kageyu Noro : Information Process of Visual Display Unit Work, Japanese Journal of Ergonomics, vol.19.No.2 (1983)
16. E. Granjean : Ergonomics in Computerized Offices, Taylor & Francis London (1987)
17. Steelcase General Catalogue, Steelcase Inc.

Human Models and Data in the Ubiquitous Information Infrastructure

Frank Berkers, Jan Goossenaerts, Dieter Hammer, and Hans Wortmann

Eindhoven University of Technology, Faculty of Technology Management
POBox 513, 5600 MB Eindhoven, The Netherlands

J.B.M.Goossenaerts@tm.tue.nl

Abstract. In this paper we address the modelling of humans in a context of ubiquitous information services. First we present the apparently conflicting requirements of, on the one hand avoiding data inconsistencies while reducing burdensome aspects of information technologies, and on the other hand, the respect for a person's privacy. These problems are analysed in the context of current ICT developments towards a ubiquitous information infrastructure. The architecture of a Human Data Manager Service (HDMS) is proposed and illustrated. It offers a solution to information modelling problems that originate from the current shift in the scope of information systems, from the enterprise to the enterprise network in the market.

1 Introduction

This paper contributes results of a project that is focussed on attaining flexibility in the cooperation of enterprises by explicitly modelling concepts that are not specific to one enterprise. Firstly we looked at human models and related data in the context of Virtual Enterprises. The lessons learned from this analysis have a wider relevance, also regarding human factors in relation to the working environment and the organization that is increasingly being supported by networked computers and information systems.

An important problem is the risk of inconsistency by the duplication of data by technically separated applications in an integrated business environment within which any person interacts with an increasing number of information systems. Assuming the connectivity of a ubiquitous information infrastructure, there exist different solutions to this problem. A forthcoming requirement then is the respect of one's privacy and related data protection. We seek an architecture that supports consistent management of human data with consideration of data protection requirements to respect one's privacy.

This paper first explains some of the key characteristics of human models in enterprise modelling and human data management. It then proposes a Human Data

Management Service (HDMS) which is neutral as regards ethical requirements and desirable rules for data protection, but which can be deployed according to principles and rules that are agreed in the relevant community.

In Section 2 of the paper we describe the current situation in which most enterprise information systems have their own human model, giving rise to significant duplication of data. It then becomes difficult to avoid inconsistencies. There is a trade-off between global availability of data and consistency with risks for privacy and excessive administrative burden. A fear of intrusion in the private life of citizens poses a justified hurdle to global connectivity and ubiquitous computing. In many countries, there are laws prohibiting combining large datasets, such as tax information and bank account information. In Section 3, we present a distributed architecture that allows for keeping data on humans consistent, and also gives support for the sharing of data in line with data protection rules. A condition is that there is extensive ontology sharing and that all data is held by and shared among well-behaved contexts. A basic HDMS architecture builds on ubiquitous connectivity for “on-the-fly” collecting of data for any application, and it assumes that only own data can be made persistent in the context’s data processing and storage device. The latter assumption distinguishes the HDMS architecture from current practice. The basic architecture is extended with replication and a publisher/subscriber pattern for working in a less connected information infrastructure, in order to improve availability of data. Data which is owned in another context, with which an agreed relationship exists, e.g. a labour contract or a citizenship, may be replicated on a persistent document, also called a hatch, in accordance with model subsidence and the data protection rules applicable for the contexts, models and data involved. A context data manager (CDM) publishes data for other subscribing contexts holding replicas of its proprietary data, and it subscribes to data from other contexts for local use. In Section 4, the HDMS is illustrated using the XML schema notation and some aspects regarding the practical deployment of the architecture are briefly addressed.

2 From Information Islands to Ubiquitous Connectivity

2.1 Problem Sketch

If one realises how often a person is registered in an organisation it becomes apparent that a change in a person’s details should not occur too often. A look at the registration of a Ph. D. student at the Eindhoven University of Technology (TU/e) sketches the problem.

The example in Figure 1 illustrates the problem of data-inconsistency for distributed data, especially in case of a change. Each registration characterises a specific human by a number of details (for example by name and address), and is based on an underlying “human model”, which is specific for an application. It is burdensome to keep this large collection of human models and model instances consistent. Think of the changes in the Ph.D. student’s details when he moves house to another city and fails to update one of the many registrations.

John Smith is a Ph. D. student at the TU/e. At the university, employee contracts are registered at the faculty, in this case Technology Management, although the Ph. D. student is an employee of the university. He is registered with social security number and bank account number on the payroll of the university. His internal phone number and office address are in the faculty phone book. The capacity group, Information & Technology, of which he is a member, also holds his contact data including his private address. He is a teacher in course 1R620. He is also registered as a sports cardholder to be able to play squash and swim with his colleagues. Some of his details are recorded in the global address book of the MS exchange server. For system access he is registered in the Novell system and in the MS Windows NT domain. Handing out of business cards spreads some more details.

The registered data consist for example of the following attributes: name, address(es), phone number(s), date of birth, employee identification, etc.

Fig. 1. Problem example

Analysing the example we note the following:

1. *Many model instances are used to model the same human being.* The same entity is modelled in different application areas, giving rise to significant overlap in the representations [8].
2. *Every model (instance) serves a specific purpose within a well-defined scope.* The model of the payroll in a company is designed to be able to pay the employee; therefore it holds the bank account number.
3. *Several model instances (representing the same entity) are present in the same business context.* All these model instances exist in the TU/e and are used in a number of applications. A change in a model instance, or a change concerning the entity that is modelled does not automatically cause an alert to update other model instances that model the same entity.

That so many applications manage their own human models and data is due to the fact that most applications have been designed for traditional technology with poor or no connectivity and without consideration of the external business context, only to serve a local function.

2.2 Predominant Human Models

In enterprise modelling three types of human models predominate. These are 1) human as a resource; 2) human as an organisational unit; and 3) human as a system user. The first two are well described in literature on enterprise modelling frameworks and on workflow management [5,6,11,12]. The third type has become more and more relevant, since usage of systems must be authorized.

Humans are often modelled as a *resource*, since they play an important role in many operational processes. In most enterprise information systems process planning functionality is present. Therefore we need to know capabilities, (qualifications, skills and roles for humans) and schedules for all resources (both machines and humans). In the example this resource model is applicable to the Ph.D. student as a teacher in an undergraduate course. The schedule that defines his availability as a (teaching) resource has to be matched with the course schedule. A human model that models the

human as an *organisation unit* basically describes the (hierarchical) relations between humans in an enterprise. For structuring organisation units into larger entities, concepts such as organisation cell have been introduced. As an *organisation unit* a human is able to assume responsibility and authority [6]. In the example these concepts apply to the Ph.D. student as a member of a capacity group in a faculty in the university.

The information system (IS) of an enterprise is nowadays composed of several applications, which are more or less integrated, by databases, web connections, object request brokers (ORBs) etc. It used to be composed of independent applications, each developed for a different business function. Even a highly integrated system as an ERP (enterprise resource planning) system does not stand on its own. The enterprise uses also office applications etc. Most of these information systems also model the human as *user*. Humans appear as writers of documents, senders of e-mails, creators of objects, reviewers of design documents and makers of phone calls etc. The actions of a system user are registered in a number of places. Registration is not the only purpose of the user model. We want to ensure or prevent some actions to be taken, therefore we use rights assigned to the user model instance or the user group (authentication and authorisation).

2.3 Shift of Scope for Enterprise Information Systems

In the relation of human models with information systems for enterprises, some shifts are taking place that also influence the development of information systems. For information system development, the area under consideration, used to be only the enterprise, whereas it is now shifting to include also the surrounding area, which we term the *market*. The market includes customers, other enterprises, public agencies, legislation, jurisdiction, environment etc. The scope of a system is the whole area which it deals with, or which may be influenced by its actions. The scope of a model is the whole area that is influenced by its definitions.

	Enterprise scope	Market scope
1	Single-site enterprises	Multi-site enterprises, business related enterprises
2	Single-site information system	Multi-site information systems, interacting or connected
3	One human-enterprise relation	Many human-enterprise relations
4	Human-human interaction	Also Computer-computer interaction

Fig. 2. Comparison of old and new assumptions on enterprises and their information systems

The summarizing table in Figure 2 is explained further. In the early days enterprises were usually situated at one location. Nowadays a single enterprise may be dispersed all over the world, and may interact with enterprises, people and institutions all over the world. The information systems developed were by technology limited to one location (for example mainframes) and designed for use by one enterprise. In contrast, technology now permits information systems to be distributed, and to be used by several enterprises. An example is tracking & tracing of postal packages using a web application. Information systems are no longer designed to operate in isolation only, they are becoming interoperable. A similar trend is seen for the human

who used to be employed by one enterprise within which he was involved in a small number of tasks. Nowadays humans may interact with the information systems of several enterprises and they may be active in many roles in multiple enterprises. Consider for example consultants who work for numerous enterprises, or the human as customer of many enterprises. Also within a single enterprise an increasing range of tasks may be allocated to the same person. Finally, also the interaction among enterprises is extended from human-to-human, often supported by paper documents, to include also computer to computer. Whereas information systems were not connected and for example orders had to be retyped to enter the system, at present an increasing number of interactions is established electronically without a need for human intervention.

The shift of scope for enterprise information systems gives rise to a review of the requirements for the modelling and sharing of data on humans.

2.4 Requirements for a HDMS

The human data management service (HDMS) is a hypothetical software service that ensures data consistency of distributed human data model instances in a specific, explicit context. The requirements for the HDMS architecture are derived from an analysis of the problems listed in Section 2.1, in a context where new information and communications technology provokes a shift of the scope of information systems, from the enterprise to the market.

2.4.1 Consistency of Model Instances

Consistent means “not self-contradictory” or “in harmony between statements”. In other words the elements of the model instances should not contradict each other. For example the registered visiting addresses should be equal. We term this *instance* or *value consistency*. On a more abstract level also the models, according to which model instances are instantiated, should not contradict. For example they should agree on the number and the types of addresses a human can have (e.g. a visiting and a mail address), what attributes compose an address and what they mean. We term this *model consistency*. Inconsistencies of instance may occur when different model instances, each in a different context, refer to the same person. We assume that the elements that are shared between different contexts are defined in the same ontology. This consistency requirement coincides with the requirement of ‘consistency and completeness’ as posed in [11]. As the scope of information systems shifts from the enterprise to the market, it becomes more difficult to realize and maintain consistency, because both the number of information systems and their users increase, and because there is no authority guarding the quality of market-wide models and model instances.

2.4.2 Broad and Narrow Scoped Models

A single model should not capture logically separable information, and it should be fit for reuse. A human model instance that has a market-wide scope should for example not include rights and duties that are transferred to that human on the basis of the position that he fulfils in the more narrow scope of an organisation, since the human

only has these powers if he ‘enters’ the enterprise context. Here, the human fulfils the position and the position holds a set of rights and duties. These relations can (and should) be modelled separately. Model instances in different contexts, e.g. the market and the organisation, that refer to the same human must be related in order to highlight what is modelled and where it is recorded, and in order to coordinate changes. The human model applied in a market has a broad scope but lists few attributes. In contrast the model used in an organisation has a more narrow scope but usually is more extensive. Think of extra tabs in a property sheet that are added for use in a more narrow scope. An enterprise information system may require a model instance with or without the ‘resource tab’. Between models with broad and narrow scope there may exist *subsidence* relationships. A narrow scoped model subsides, or settles lower down, for a broad scoped model, and the latter is said to supersede parts of the narrow scoped model. A change in a superseding model also affects the subsiding model’s instances.

2.4.3 Scope Sensitive Data Management

As many model instances model the same person, we conclude that different applications, users, and enterprises, each in its context, stores similar data. Scopes and contexts may be nested in encompassing scopes or contexts. For example a person appearing in the different contexts of a hospital, as a patient, and at work, as an employee. At present, information systems in both contexts would record data on (the same) person, with disregard for the opportunity to share models and data. In the case that a broad scope human model exists in the country, then both the hospital and the company could prefer, or they could be obliged, to define their narrow scope human models subordinate to the country’s wide-scope model. For attributes such as name, address and birth date, the company’s human model would then be subordinate to the country’s human model, but for attributes such as capabilities, office, wage, salaries paid, etc, it would be *leading*, as the company is the scope for and owner of these data. Data could then be stored at the context where the model is leading, and data of superseding models, should be requested from the encompassing contexts. Applying this approach to the extreme would lead to zero-replication and hundred percent consistency of the data, with the consequence that a single update is made for every relevant event, such as the person moving to a new house. This approach would require connectivity among all information systems storing data affected by the relevant events, whenever they may occur.

2.4.4 Data Protection and Privacy

Without making precise and firm statements about the level of protection of data and privacy that is suitable for the workplace, in the relation between employee and employer, between customer and supplier, between a person and the public, it is clear that the increasing connectivity calls for a comprehensive approach to this requirement. Whereas “the right of everyone to respect for private and family life, his home and his correspondence,” was expressed in the European Convention on Human Rights in 1950 [9], it was only back in 1978 concluded by the Lindop Committee in the UK that “the function of data protection law should be different to that of a law of privacy; rather than establishing rights it should provide a framework for finding a balance between the interests of the individual, the data user and the community at

large” (Cited by [9]). Because legislation runs far behind when compared with the possibilities created by information and communication technologies, it is appropriate to require the technology to provide a framework on the basis of which concrete data protection measures, proposed by a community, can be implemented. Seen from the perspective of human rights, and given the trend towards ubiquitous information infrastructure, wide-scoped human data management may well provide a viable alternative to the human models and data models in current narrow-scoped and database centric information systems. Current application developments disregard opportunities of data sharing outside the enterprise and have no coherent answer to the growing risks of violation of data protection law. New risks emerge from the technology trends, in combination with the current lack of a framework for balancing the interests of the individual, the enterprise and the community at large.

3 Architecture of the HDMS

This section presents the concepts needed, and the architecture of a human data management service (HDMS) that meets the requirements listed in the previous section. The architecture is illustrated in the next section.

3.1 Concepts

The architecture is built using the following concepts: context, the role/actor pattern, and the data dependencies subsidence and superseding.

As we described in the analysis, every model (instance) serves a specific purpose. This purpose prevails in a specific ‘area of application’, which when considered as a whole, is called the scope of the model. For a given model (instance) the term *context* is used to refer to the models that immediately surround it and help to define or clarify its meaning.. Drawing on the meaning of context in linguistics, Mills et al.[7] define context as a situation of human/computer interaction. Stamper in [10] defines it as “*a norm system of semantics and pragmatics, and rules of the social group the interpretant belongs to.*” Whereas the latter definition addresses a wide scope context of the interpretant, the HDMS architecture extends Mills’ definition of context to include also situations of enterprise/computer interaction. In the domain of enterprises and persons interacting with computers, data protection rules will limit the flow of data between the contexts pertaining to the individuals. Hence, the HDMS architecture defines the *context* as a situation of human or enterprise/computer interaction for which specific rules can be defined to control the in-flow, out-flow and storage or persistence of data, and for which data processing services are available. The HDMS contexts are defined in the so-called “information domain” or “cybernetic domain” which must be distinguished from the physical domain [4]. Typical contexts are the enterprise context within which data on products, services, customers and employees may be recorded, and also the private context, including the data about one self.

For enterprise and organisation contexts the role/actor pattern [2] is relevant. The *role/actor* pattern is used when a role can be played by *one or more* actors, or when

rules on the roles must be expressed independent of the actors, or when actors must be *matched* to roles. An example of a matching rule is that the lecturer of course 1R620 must be graduated. An example of a rule is that the lecturer of a course must not be a student in the same course. The department (enterprise context) defines who can be a lecturer and what he must do; the department defines the role of lecturer.

Role definitions imply requirements on the actor. The requirements of the role must be met by the “data” in the private context of the person who fills the role, and thus by his human model. The predominant human models define general roles that a human can perform, i.e. a human in a specific context is likely to be modelled ‘in’ a specific role, a specialisation of one of these general roles. The organisation unit role is always played in the context of an organisation, the resource role is defined for an activity, the system user role is defined for an information system (application).

For each context there is a distinction between the data that is stored within the context and for which the context is responsible, and data that must be retrieved from another context. If context A depends on context B for a specific piece of data that originates in context B, then we speak of a data *subsidence* relationship. For example names are used in the university, but are given in a family context. The given name of a person will supersede any name that can be used in a company context. On the other hand the university defines employee numbers for its internal administration.

With the role/actor pattern and the concepts context and data subsidence we can describe the problem of human data management as a problem of managing the consistency and protection of data models on humans in different roles in different contexts, subject to data subsidence and (norms and) rules, e.g., on access to and exchange of data, defined in and among the contexts.

3.2 Conceptual Architecture

In the conceptual architecture we describe the HDMS in two phases. First a basic service is presented in which very strict rules apply on the storage of data within any context: only data that is not subsiding to other data can be stored in the context. In a second extended service this rule is relaxed to allow for replication and publisher-subscriber relationships between contexts. For both architectures, we address the situation that a private context is temporarily bound to an organisation context in accordance with a role/actor pattern.

3.2.1 A Basic Human Data Management Service

We assume that in every context someone (or something) has the responsibility to provide the context with proper data. In (the context of) an information system this is usually ensured within the application. We make the role of a Context Data Manager (CDM) explicit for every context. The main task of the CDM is to ensure that the context is provided with the right and consistent data for its processing, and that the data from the context is only provided to other ‘authorized and trusted’ peers, as defined by data protection rules. Within a community of “peers”, such as defined in a legal system, the ownership of data can be defined. A particular peer is the proprietor of the data within its context.

The CDM has the following functions for working with its data in its interactions with other CDMs, all this within the constraints of enacted data protection rules.

1. *Store the own data.* Data for which the context is the proprietor must be stored for making it available in the future (persistence).
2. *Express requests for external data.* Data that is needed within the context but not available must be requested from elsewhere, upon receipt it must be processed but it cannot be stored or duplicated to prevent violations of the data protection rules. This situation typically occurs when an organisation's human model and data is subsiding to the "public" human model, and related data. The CDM must be able to find the contexts it depends on.
3. *Answer external request for the own data.* The requested data will be passed on to contexts that can be trusted as regards their abiding by the data protection rules. A typical example is the request from a company context to a private context; both contexts adhere to a national model and can therefore be trusted.

In the proposed architecture all data elements have a unique context of origin and persistence. Combined with ubiquitous access this allows us to achieve consistency: every model is composed of references to the context of origin of a specific data element.

The private context. We propose a unique 'token' for each human; this is a 'minimal' model instance that represents a specific human (in the private context) according to a "national-scale" human model. This 'token', we term it the *p-sign*, can be bound in other contexts in which the person fulfils a role (i.e. data on this human is used). For this, the latter contexts will have to enact models that are subordinate to the national-scale human model. A single p-sign can be bound to role/actor patterns in different contexts.

The binding of a p-sign to a role/actor pattern in an organisation context gives rise to some form of run-time binding of contexts, as the person takes up his role in the organisation. There is a possible problem of availability of data when the organisation frequently needs data on the person, even when absent.

A lasting relationship between contexts could be supported by a "hyper text" *document* that each CDM maintains for each one of the other contexts with which it maintains an agreed relationship, such as an employment contract between an organisation and a person, or the citizenship of a person in a country. In this case, both the organisation's and the person's CDM maintain a document for each other, which we will call a *hatch*. When the organisation's CDM needs certain data on a person it will first look for his hatch, if there is none because the context had no agreed upon relationship, the CDM can contact a broker. Hatches allow us to bind contexts, for instances of the person who is present in the role. When bound, all the person's data could be readily requested by and provided to the organisation. When unbound, the urgency of the request could determine whether to wait for the next presence of the person in the role, or to route the request to the person's CDM via the communication infrastructure.

Since the data subsidence relationships and other dependencies are expressed, and data can only be stored in the owning context, any CDM explicitly relies on many

other CDMs to keep data available and up-to-date at all time, as well as on a communication infrastructure to access it. This implies significant availability risks and plenty dependency relationships.

3.2.2 Replication and the Publisher/Subscriber Pattern

In addition to availability risks, supporting the relationships between contexts by means of hatch documents and without replicating any data puts a significant demand and workload on the communication infrastructure. Moreover, certain data is unlikely to change very often, and therefore does not justify frequent communication costs, however tiny, whenever it is needed. Instead of accessing the elements, as they must be processed, it is also possible to copy (replicating or caching) certain elements, with observation of the data protection rules that apply for these elements. But replication also calls for an updating mechanism to avoid data-inconsistencies: those who have replicated the data must be notified of a change of the data by the owning CDM. The publisher/subscriber pattern [3] is suitable for this case. An alternative mechanism is that a CDM checks on changes by periodically polling the owners of the data.

Reuse and consistency of models and data is facilitated by CDMs that use published models for their data. For example all CDMs for the private contexts can publish according to a centrally managed human data model. CDMs may express a data subsidence relationship with the CDM that publishes the model. The company CDM manages a number of replicas of model instances of its employees, by expressing subsidence to the country's human model, and by subscribing to the private context of each employee. The latter context offers a publishing service for selected hatch documents, e.g. for their employers and newspaper subscriptions.

3.2.3 Data Protection at Work

The CDM must not only control what data to share, but also with whom. We explore the application of the contexts and their binding for a simple scenario. Access to particular data about a person working in a role at a company is restricted by several data protection rules or regulations. Several laws are applicable to registration or use of personal data. For example, it is not allowed to record and publish a person's health condition, but it may be allowed to publish the aggregate health condition of the work force in a certain industrial sector. On the other hand, the interest of a physician in the ontogenic development of the patient may cause an interest for certain data about the patient's individual work postures. How to meet all data requirements without violating the data protection rules? The answer is based on selective data-collection by both the private and company contexts during the periods of engagement, and the erasing of confidential data upon disengagement. Each party collects data on the work aspects that matter for its own ontogenic development, or for related feedback loops that have been enacted observing data protection law. But, as the worker ends his working day, his private context splits from the company's context and all "confidential" traces as to whom performed which work and how, could be erased. In the same way, all confidential traces of the work performed can be erased from the person's private context. However, the anonymous data on the work process may be stored for analysis and action, for instance by the company's quality engineer or health service, or by the worker's physician.

4 Illustration and Discussion

4.1 Illustration

We return to the example of the sports administration to illustrate the context data manager. In Figure 3, we assume the existence of a national human model defined in an XML schema “HumanModel1.xsd” with elements name, address and birth date. For doctoral students there is a contract model, defined in the TU/e repository, in the schema “ContractModel1.xsd”. Furthermore, the sports administration rules specify that students have access to all facilities. For employees an employment contract is required and they have access to a limited set of facilities, from which they must chose. Notwithstanding Ph.D. students being employees, they have access to all facilities. The document instance in Figure 3 is the hatch of John Smith in the context of the information system of the sports administration of the TU/e. The hatch manages the data and data references for John Smith.

```
<?xml version="1.0"?>
<HatchJohnSmith
  xmlns:h="http://www.nat_rep.nl/namespaces/humanmod"
  xmlns:c="http://www.TUE_rep.nl/namespaces/contractmod"
  xmlns:sa="http://www.TUE_rep.nl/namespaces/sportsadmin"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xsi:schemaLocation=
    "http://www.nat_rep.nl/namespaces/humanmod HumanModel1.xsd
    http://www.TUE_rep.nl/namespaces/contractmod ContractModel1.xsd
    http://www.TUE_rep.nl/namespaces/sportsadmin Member1.xsd ">
  <h:Psign ref="www.humanmodel.nl/JohnSmith"/>
  <c:Contract ref="www.tue.nl/personaladmin/JohnSmith"/>
  <sa:Sportscard_nr> 1102494 </sa:Sportscard_nr>
  <sa:Cardtype> Ph.D. </sa:Cardtype>
  <sa:Facilities> all </sa:Facilities>
</HatchJohnSmith>
```

Fig. 3. A document instance in the sportsadmin system

This document is tailored to the role of John Smith as a sports facility user. The model sources are assumed to be implemented as XML schemas and available in suitable repositories. This model instance extends both the private human model in the national repository and the contract model in the TUE context, according to the zero-replication HDMS architecture. Data on John Smith in his private context, and in his contract hatch at the personal administration can not be stored in this document, but must be requested from the referred documents, as indicated with the “ref” attributes of the empty elements “Psign” and “Contract”. Each element has either a value, when the context is the owner, or a reference, when the element is owned by another context. Or with other words, if an element has a value this means that the data originates from this context. The sports administration context has a hatch document for every member.

The HDMS with replication is illustrated by introducing some small modifications in the example in Figure 3. The empty “Psign” for the private context could be replaced by an element that subscribes to the referred document, in this case the

context at “www.humanmodel.nl/JohnSmith.” In the subscribing “Psign_S” element, the elements of “HumanModel1.xsd” are listed to hold the replicated data (Figure 4).

```
<h:Psign_S publisher= "www.humanmodel.nl/JohnSmith">
<h:Name> John Smith </h:Name>
<h:Date of birth > February 29, 1960 </h:Date of birth>
<h:Address> ..... </h:Address>
...
</h:Psign_S>
```

Fig. 4. An element subscribing to a human model instance

The context of John Smith now has to keep track of the hatches that subscribe to changes of its data, such as his address, by generating “publish events” on changes of these elements.

A final issue that we address is the nesting of contexts and the implied opportunities to gain efficiency by centralising the efforts of data management. The sports administration system is within the TU/e context. This larger context will for sure have a model for this person, since it is either an employee or a student of the university. It is recommended that the TU/e CDM takes care of the data that subsides to data in the private context. The responsibility to provide the sports administration with data is then shifted to the TU/e CDM. This is possible when there’s a strong link between the contexts, which is the case: the sports administration system and its users, the employees of the sports centre are in the context of the TU/e, the users are employees of the TU/e. The result is that in the document of Figure 3, the references to the human model and the “Psign” of John Smith are removed. Instead, the sports administration will rely on the contract model and the contract model hatch of John Smith in the personal administration of the TU/e. The address of the member with sportcard number 1102494 is now looked up via the personal administration context, which will, as the only context within the TU/e, subscribe to John Smith’s “Psign”. Hence, the number of subscribers to the person’s context is reduced by the number of nested contexts of the TU/e that previously would have subscribed to this element. The TU/e context subscribes on behalf of the nested contexts.

4.2 Discussion

So far we have addressed requirements, independent of actual implementations of ubiquitous information infrastructures. Regarding the implementation of private contexts we envisage the use of mobile devices for holding private contexts in combination with “national-scale” authentication support for access to resources [1]. Although the rapid development and implementation of the HDMS architecture may seem rather utopian, a nation-wide or company wide migration towards an infrastructure that would feature HDMS services could be accelerated by drivers related to the following discussion points: the need for fine-grain data protection, the extension of the architecture to other entities, the hosting of contexts, and the ICT development economies inherent in the sharing of models. Each of these points is briefly addressed. Regarding the granularity of data protection, the example in section 3.2.3 makes a distinction between the ontogenic development of the human for which

privacy matters, and the roles executed in the working environment for which purpose-specific feedback loops could be enacted.

Although we did take into account specific characteristics of the human in proposing the HDMS architecture, the proposed architecture might be applicable to other modelled entities such as high value products and capital goods. Products do not actively play a role, but we can define their ontogenic development and information system support for it [4].

Smart mobile devices might deliver the private context services on which HDMS architectures rely for their data protection. For persons who can not afford or do not want to use such devices, *hosting services* could be created which could extend records in municipal administrations, such as the birth certificate to which any persons' life in an administrative infrastructure is usually traced back. Hosting services are also possible for companies and public agencies, of course. The pervasive nature of the HDMS makes it virtually mandatory that public agencies or similar trusted bodies assume a role in disciplining the service providers, such as is the case for banking. It can be noted that certain hosting services are already offered for simple models and model instances. Examples are phone companies that publish addresses and phone number, and job intermediaries that publish anonymous CVs. Numerous intermediaries store essentially the same data.

A final discussion point concerns the development economies inherent in the sharing of models. Indeed, if the same model (instance) is used in a number of contexts it may as well be managed by a doming context. This principle could be called the *subsidiarity principle for information modelling*. Any model should be provided at the level at which its reuse is widest. In general, a doming context/CDM should be introduced if the advantages of reuse outweigh its disadvantages.

5 Conclusion and Future Work

Combining the requirements behind the predominant human models in enterprise modelling with the requirement for fine-grained data protection rules, we have specified a first version of a Human Data Management Service that depends on the connectivity offered by the anticipated ubiquitous information infrastructure. Two variants of the conceptual architecture were proposed: a basic zero-replication architecture posing high demands on the communication infrastructure, and a less demanding extended variant of the basic architecture with replication and a publisher/subscriber service linking autonomous and nested contexts. Both architectures have been described by means of the concepts of scope, context, the role/actor pattern and hatch documents for representing agreed-upon relationships between peers. The XML schema definition language and XML were applied in the brief illustrations of both architectures.

Further work should address several issues in far more detail. These issues include: the structures and nesting of contexts, as they occur in organisational hierarchies and in communities of citizens and other legal entities with "equal rights"; the application of this architecture to other entities, the interfaces with and extension to other entities, i.e., resources, products[7] and enterprises; the enactment of data protection rules in communities; the alignment with e-commerce standard proposals such as ebXML, regarding the publishing of models; security issues; and version management for models (node versioning).

Acknowledgements. The authors thank Freek Erens and Theo Kranendonk from the Baan Development Nucleus team for their contribution to the clarification of the concept of scope. The second author thanks John J. Mills from The University of Texas at Arlington for his contribution through joint work on context modelling.

References

1. Butler Von Welch, R., Engert, D., Foster, I., Tuecke, S. and Kesselman, C.: A National-Scale Authentication Infrastructure, *Computer*, December 2000, 60-66
2. Erikson, H.-E., and Penker, M.: *Business Modelling with UML - Business Patterns at work*. OMG Press, John Wiley & Sons, New York (2000)
3. Gamma, E., Helm, R., and Johnson, R.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley (1995)
4. Goossenaerts, J.B.M.: Industrial semiosis – founding the deployment of the ubiquitous information infrastructure. *Computers in Industry* 43 (2000) 189-201
5. Kosanke, K., Nell, J.G.: Standardisation in ISO for enterprise engineering and integration. *Computers in Industry* 40(2-3) (1999) 311-319
6. Kosanke, K., Vernadat, F.B., Zelm, M.: CIMOSA: enterprise engineering and integration. *Computers in Industry* 40(2-3) (1999) 83-97
7. Mills, John J., Goossenaerts, J.B.M., Pels, H.J.: The Role of Context in the Product Realisation Process. *Proc. CIRP Design Seminar*, Stockholm, June (2001) 175-180
8. Scheer, A.-W.: *Business Process Engineering -- Reference Models for Industrial Enterprises*. Springer Verlag, Berlin (1994)
9. Slee, D.: The Data Protection Bill 1998: a comparative examination. *Information & Communications Technology Law* 8(1) (1999) 71-109
10. Stamper, R.K.: Signs, organisations, norms and information systems, *Proc. Third Australian Conf. on Information Systems*, Univ. of Wollongong, Australia (1992)
11. Vernadat, F.B.: *Enterprise Modelling and Integration, Principles and Applications*, Chapman & Hall, London, England (1996)
12. Workflow Management Coalition Terminology & Glossary 3.0, WfMC, (1999)

Motion Simulation of the Human Workers for the Integrated Computer-Aided Manufacturing Process Simulation Based on Info-Ergonomics Concept

Sayaka Imai¹, Takashi Tomii², and Hiroshi Arisawa²

¹ Department of Computer Science, Gunma University,
1-5-1 Tenjin-cho, Kiryu, Gunma 376-8515 JAPAN
`sayaka@cs.gunma-u.ac.jp`

² Graduate School of Environment and Information Sciences,
Yokohama National University,
79-7 Tokiwadai, Hodogaya-ku, Yokohama, 240-8501 JAPAN
`{arisawa,tommy}@ynu.ac.jp`

Abstract. This paper presents modeling of Working Process and Working Simulation in factory works. Also a concept for Mediator-based human body/motion modelin for application to Info-Ergonomics is offered. In Manufacturing Process Design, many simulations are executed in order to improve efficiency of designer's work. But in many cases they are focusing on simulating and evaluating machines' performance. In the present paper we will propose Info-Ergonomics, and Working Simulation by using CG. Also, we consider it as a possible way for unifying all the data used in various applications (CAD/CAM, etc) during the design process and evaluating all subsystems in a virtual Factory.

1 Introduction

As a result of the significant advancement in computer technology many new applications of image and graphical data are brought to life. Those include the fields of Image processing and retrieval, Engineering Simulation, Tele-robotics, and so on. Computer graphics (CG) in particular has been widely used for a long time in CAD/CAM/CAE systems as a visualization tool. Recently some new applications supporting the designing process of a virtual factory and simulating its work are gaining popularity in areas like process planning and work scheduling. In addition to the CG, they involve many new other types of data, such as video and X-ray images, sensing information, etc. High complexity of those information causes a lot of difficulties to the researchers and developers. Here we would like to point out two problems. We consider important for the future development of those systems. First, as those applications use mainly application-oriented data, sharing data with other applications is quite difficult and very often —impossible. In our opinion, a flexible automation system requires data about a given product to be used in all applications.

Second, we think that eco-factory human factors are usually overlooked or underestimated. Human labor has an important role in the manufacturing process. It is examined from the viewpoint of industrial engineering, ergonomics, etc. and results are used in product planning, working analysis, and work environment design. But aspects like comfort of the workers and optimal human-machine co-existence are not explored.

As a promising solution to the above problems we have proposed the use of Real World Database(RWDB) and Info-Ergonomics simulation[1]. RWDB is able to capture various types of data, namely characters, video images, 3D graphics, and shapes of 3D objects. We call all those *Real world data*. The data of all types are unified and stored in a Multimedia Database(MMDB).

On the other hand, we consider it necessary to focus on human-machine cooperation, especially for employees in the factories and analyze their work evaluation, environment, and amenities against this background. We offer a model for analyzing employee's work evaluation and using the results in the manufacturing process design. We consider it possible to use Info-Ergonomics simulation data in human-machine cooperation simulation and in the production design, which would result in an integrated data model for manufacturing process applications. In order to realize this application, we consider that the human body/structure and motion modeling for the human Body/Motion database(for Working simulation system) is very important. We propose a methodology for modeling of the human body and its motions in order to store and query them on a database. For storing all available data about human motion, we need a precise human body model as well as method for representing its motions.

This paper presents an extended Info-Ergonomics system and Mediator-based modeling of workers in a factory, which includes the human body structure, work motions and work semantics.

2 Info-Ergonomics

2.1 Concept of Info-Ergonomics

Info-Ergonomics(Information Ergonomics) is the concept for computer aided system, in the field of such as Ergonomics, Industrial Engineering, factory work design about factory workers and so on.

The technical bases of Info-Ergonomics are Real World Database System, Mediator-based human body/motion modeling and graphic simulation system for human motions. Details of the concept of Info-Ergonomics is shown in [2].

2.2 Info-Ergonomics for Factory Workers' Motion Simulation

Info-Ergonomics is one of the research priorities in the integrated computer-aided manufacturing process design from the viewpoint of human factors. Recently modeling the machines in the factory, creating virtual machines by using CG, simulating their work and evaluating it are coming into use gradually [3].

But modeling human beings and creating “virtual employee” by CG still has very limited use, because of the human body’s high complexity and the limits imposed by the computer techniques.

However, recently the development of computer technology enabled the modeling of complex machines based on kinematics and kinetics characteristics, analyzing machine’s motions and represent virtual machine by means of CG in real time.

We focus on human body modeling along with the machines, and evaluating employee’s working in the factory from human-machine co-operation and employee’s point of view, so we propose Info-Ergonomics as the framework for CG simulation and evaluation of virtual employees’ work.

The purpose of Info-Ergonomics can be defined as follows:

- Pursuing comfortable environment for employees (factory workers).
- Simulation and evaluation of employee’s action with simplified human body.
- Modeling of skilled employees and storing their skills into Multimedia Database.

Info-Ergonomics provides the designers with modeling, evaluating, and visualizing tools for designing the optimized working environment for the factory employees. This is an important component of “virtual factory”.

As the whole informations are schematized and integrated based on the RWDB concept, the user can extract arbitral part of them and reorganize multimedia data adapting to requirements of application.

The use of CG has significant advantages. For instance, Info-Ergonomics-based simulations allow more precise evaluation than real measurement because virtual employees can be made to do any more (even those causing pain, dangerous ones, etc) and, in addition, we can perform the evaluation itself at a much more detailed level visually.

And then, to replay the skilled employees’ work in the Real World Database (RWDB) with CG, and to study from all angles is visual aids for beginners. In addition, we can keep safety and work space for employees, and design comfortable work environment.

3 The Human Body/Motion Modeling Based on the “Mediator” Concept

3.1 Introduction of “Mediator”

In order to construct a human body/motion database, storing human body and motions, which are approximated to capturing data from real world by using 3D scanner and stereo video cameras, and spatiotemporal query method are needed. But surface data (polygons) captured by 3D scanner and spatial points sequence data analyzed from stereo video are very huge data, and are not able to display in real time. As a solution of this problem, we propose the “*Mediator*” concept about human body/structure and motion. In our previous research, we proposed

the Shape Mediator[4][5]. In this paper we extended this concept to the human body and structure, and then we propose the human Motion Mediator. Each Mediator will define as follows.

- Shape/Structure Mediator — We suppose a human body as a 3D graphic model that is constructed from “parts”(head,arm,hand,...), which don’t change its shape along time, and “joints”. Each part is defined as collection of individual polygon surface and texture, and each joint is assigned Degree of Freedom and Range of Motion.
- Motion Mediator — We suppose the human’s motions are sequence of postures, consequently we define the human’s Motion Mediator as collection of characteristic internal postures, namely, relative time and joint angle(value) of each Shape/Structure Mediator in this time, in which characteristic internal postures of a motion of a human.

Shape/Structure Mediator is made individually. Therefore, Shape/Structure Mediator reflects difference of each human shape and range of motions. On the other hand, Motion Mediator is made for each motion of a human, and represent differences each motion of a human. By using this Mediator concept, you can represent the human body’s shape, structure and motion in database. Considering Mediator, expression of a characteristic by using minimum data and of common semantics between common data and Mediator.

In next subsection, we discuss about “Hub”(Common Model) and Mediator Creation based on Hub.

3.2 Creation of Mediator from a Hub

A Hub is the information for expressing the essence in common and standardizing it about shape and structure of the human body and their motions. Definitions are given as follows.

- Shape and structure Hub — is expressed with the graphic object which has simplified the composition parts of a human body, and sticks the “semantic” information which the object has in the specific position of the surface. Furthermore, each part has “joint position” which defined connection between parts, in order to describe the structure of the whole human body.
- Motion Hub — is expressed with the intermediate posture along which it surely passes in case it appears in common with the motion is performed using the mutual position relation of the human body parts in shape and a structure Hub.

It can be said that the Hub shows the minimum “item” for describing the characteristics about shape, structure and motion of the human. The Degree of Freedom and the Range of Motion of the human are able to measure by carrying out easy motion for the human from a medical viewpoint. As opposed to the Hub, Mediator is data which transformed the Hub to the example a concrete individual exception and by motion, gives value. Next, the technique of creating Mediator from Shape/Structure Hub is follows. The Shape Mediator can be

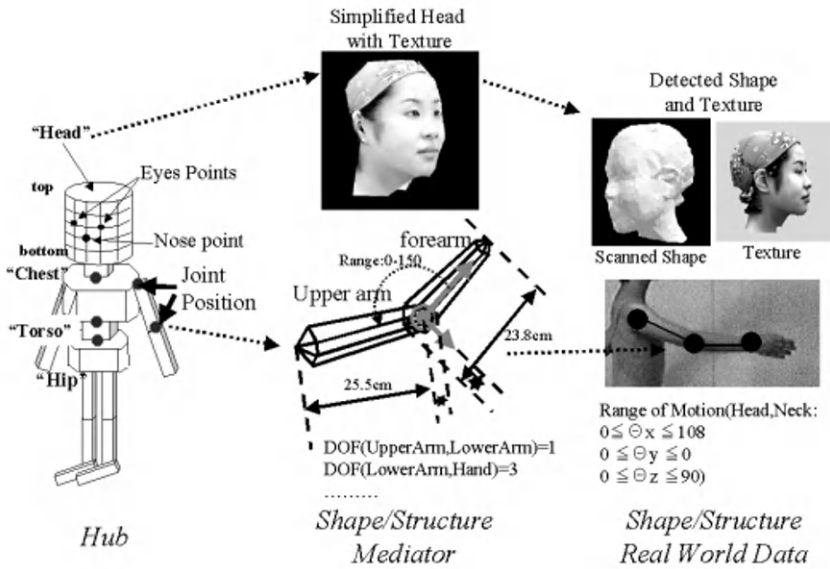


Fig. 1. Hub, Mediator and Real World Data about a shape and structure of the human

offered by transforming and carrying out fitting of the Hub using the detailed data obtained from the 3-dimensional scanner. Therefore Mediator can be constituted easily. Fig.1 shows an association of Hub–Mediator–Real World Data about shape and structure. On the other hand, when Motion Mediator is created from a Motion Hub, it is important what intermediate postures are chosen as a Motion Hub. At a present stage, the designer has to design a intermediate posture for every motion. For example, a characteristic intermediate posture sequence which generally raise “Motion which tightens a screw with a drill” in factories can be show as follows.

Tightening a screw using a drill:

I: Starting Motion – The posture in which the person is touching the drill.

II: Motion

II-1: The posture in which the person located the drill at the tip of a screw.

II-2: The posture into which the person finished tightening a screw with a drill.

II-3: Next II-1, repeat sometimes.

III: Ending Motion – The posture to which the person turned the drill.

Generally, it is very difficult to extract human’s posture precisely from videos by using stereo video cameras etc. But, if it restricts to the specific postures specified by the Motion Hub, the analysis of postures (joint angles) can be carry out easily by means of detection of corresponding points of characteristic points on the human body. Therefore, the Motion Mediator can be create from the Motion Hub. Fig.2 shows a relation of Hub–Mediator–Real World data about a motion of tightening a screw using a drill of worker in factories. The posture

sequence about Hub, Mediator and Real World Data shown in Fig.2 displays joint value data of motion for convenience as a shape of a posture.

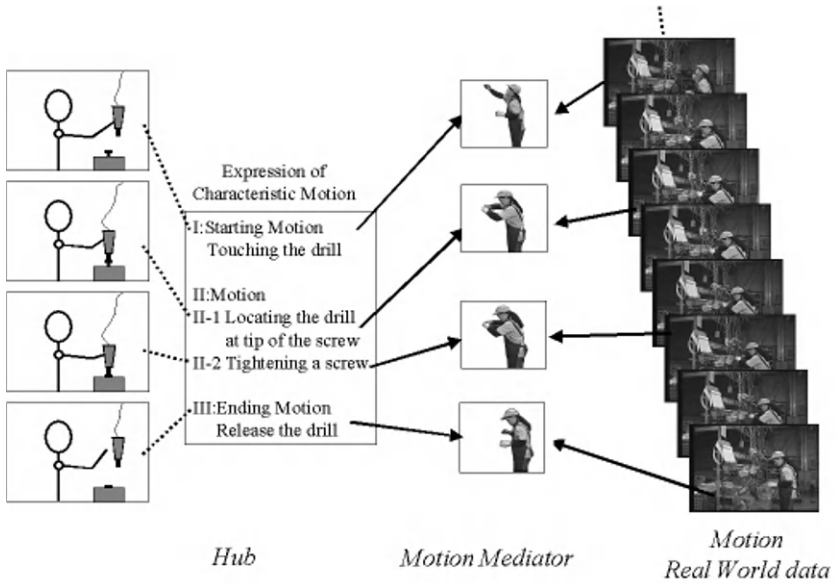


Fig. 2. Hub, Mediator and Real World Data about a working motion of the human

4 Info-Ergonomics Database Based on the Human Body/Motion Database

4.1 Objectives of Human Body/Motion Database

Objectives of Human body/motion database are follows.

1. Storing a huge number of information about human motion with their intention.
2. Expression each human motion of same human as different instances in the database.
3. Query information in the database from point of view aggregately and intensively, for example, queries which calculate average value of bend value in motions of 100 people.
4. Spatiotemporal query based on common points between human body and motions.

For the purpose of satisfying these objectives, especially 3. and 4., it is necessary to unify the character of the human body and motion in the database schema.

And it is necessary to correspond between “type” and “instance” from the point of view of 2., and to express shape and motion by using a few data as a various type of index from the point of view of 3..

In the earlier our research, about the human body parts, we offered basic common information (Hub: for example, the human head can be approximate cylinder), and *Mediator* which fitted Basic Model to real body shape(for example, surface which captured by 3D Scanner). And we discussed its ability for spatiotemporal query [5]. But for the purpose of storing the human motions, the human structure and motion have to be modeled in database schema.

4.2 Realization of a Database of Body Shape and Motion Based on Mediator

In order to accumulate motion of the human into a database, as we described previous section, the data belonging to each level of a common model, and a real world model must be able to be accumulated with the reference relation. The kinds of data are not only a character and a numerical value but a stereo image, and various things, such as a polygon texture.

Moreover, query to a database is performed, combing the data accumulated freely, and ad hoc query is performed according to user’s purpose. Meanings of Mediator, which was proposed in this paper, are not depended on data models for data expression. However, in order to realize the above query, the data model need to have the small unit of the data which defines as the mechanism with the simple reference relation of data, and is accumulated is desirable.

Then in this paper, in order to design human body, structure and motion database, we adopted functional data model “AIS(Associative Information Structure)”[6] as a schema description. Fig.3 shows database schema and its instances of the human body/motion.

For example, corresponding to operation with the real world, motion Mediator is defined, and accumulated as instances, the motion Mediator instance consists of the sequence of a intermediate posture, and the one intermediate posture is defined by the joint angle value for every joint. Thus, a complicated relation can also be directly expressed as relation of an instance level. Moreover, you can query to tracing this path freely. For example, a query, which is “Parts shape of the joint part both ends at the time of being in a certain motion”, can be realized by tracing the following route, (1)–(2)–(3)–(4)–(5) in Fig.3.

4.3 Consideration about Flexible Query of Motion

In Fig.3, the instances of Hub, Mediator and real world data, which correspond to human shape, structure and motion, are accumulated separately. Moreover, as Shape/Structure and motion of the human are independent each other, it is possible to combine a human body and motion by query variously. It is realizable not only the combination of motion of the human and the body of the human captured from real world, but to create the virtual motion animation as a query result. In short, 3-dimensional animation “Motion of the human B who imitated motion of the human A” can be create by querying “Motion of the human A”

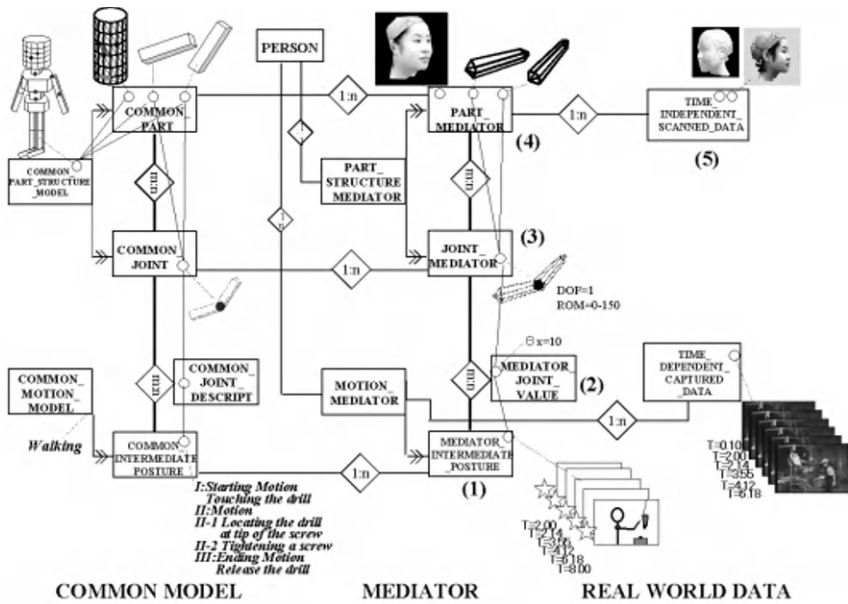


Fig. 3. Database Schema and its instances of the human body/motion

and “The body of the human B” and combining them. Mediator expresses parts shape and structure, and the individual feature of motion and has also described the difference between same parts shape and structure, and motion. Therefore, various query can be performed from Motion Mediator, for example, “An mean time of tightening a screw in motions of worker A”, “An average of unit work time”, and “Maximum of the distance which extend the worker A’s right arm to take a drill”.

Moreover, in Fig.3, The structure allows powerful and flexible data retrieval. As a most interesting type of query we consider the reconstructing of a new world using CG data from the database. Generally, CG Data includes shapes, polygons, textures, positions, lights, view points, motions and so on. Thus, 3-Dimensional CG data is huge and extremely complex. In the Info-Ergonomics Database, the query result is provided as complex data object. And you can get working simulation data as a CG by reconstructing it.

5 Info-Ergonomics Simulation

5.1 Info-Ergonomics Simulation System

We have designed an Info-Ergonomics Simulation System based on RWDB and ENVISION(DELMI A Corp.)[7], and are developing now a prototype system. Fig.4 shows the Architecture of Info-Ergonomics Simulation System. Basically working simulation is designed as a part of man-machine coexisting system, and

then evaluated. All results can be accumulated into the Multimedia Database together with worker's data. Those data might be referenced by the Manufacturing Process designers when designing a new process or a factory. In short, we propose Info-Ergonomics as an approach supporting the work analysis and work design with knowledge and data.

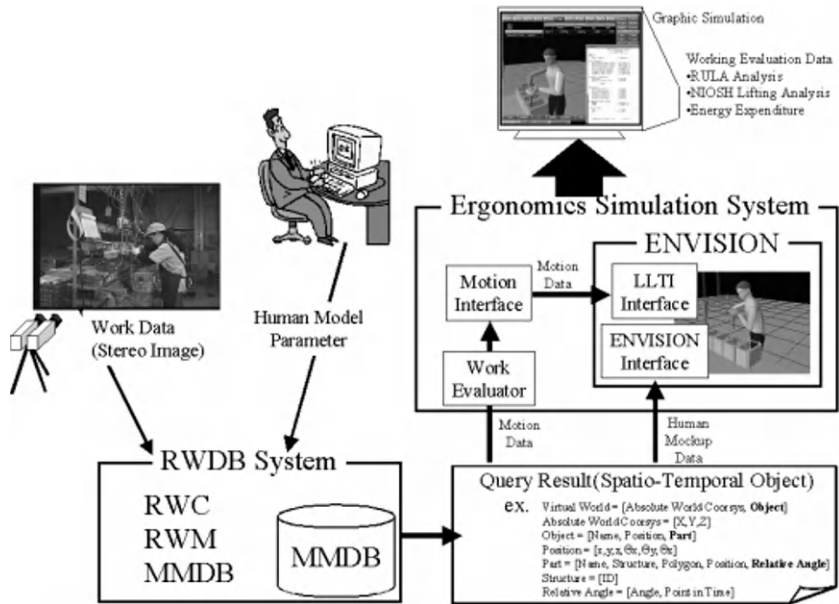


Fig. 4. Architecture of Info-Ergonomics Simulation System

6 Concluding Remarks

In the present paper we have reported the concept of Info-Ergonomics as a framework for Manufacturing Process Design. It supposes shared data usage between applications by means of *Real World Database* which can incorporate all the data through an integrated data model. Such approach would be beneficial to Manufacturing Process Design. Future work should focus on the detailed design of the *Real World Database System* and the further development of the simulation system.

Acknowledgement. A part of this research is supported by the IMS-HUMACS project Japan. This work is also supported in part by the Grant-in-Aid for Encouragement of Young Scientists from the Ministry of Education, Culture, Sports, Science and Technology. (No.13780198)

References

1. S. Imai, K. Salev, T. Tomii and H. Arisawa: Modelling of Working Processes and Working Simulation based on Info-Ergonomics and Real World Database Concept. Proc. of 1998 Japan-U.S.A Symposium on Flexible Automation(JUSFA'98) (1998) 147–154
2. H. Arisawa, T. Sato and T. Tomii: Human-Body Motion Simulation Using Bone-based Human Model and Construction of Motion Database. International Workshop on Conceptual Modeling of Human Organizational Social Aspects of Manufacturing Activities HUMACS 2001(2001) (to appear)
3. K. Noro: Illustrated Ergonomics. JIS (1990)
4. H. Arisawa and S. Imai: Mediator-based modeling of factory workers and their motions in the framework of Info-Ergonomics. Human Friendly Mechatronics(ICMA 2000), E.Arai, T.Arai and M.Takano Eds., , Elsevier Science B.V. (2001) 395–400
5. T. Tomii, S. Varga, S. Imai and H. Arisawa: Design of Video Scene Databases with Mapping to Virtual CG Space. Proc. of the 1999 IEEE International Conference on Multimedia Computing & Systems ICMCS'99 (1999) 741–746
6. H. Arisawa, T. Tomii, H.Yui, H. Ishikawa: Data Model and Architecture of Multimedia Database for Engineering Applications. IEICE Trans. Inf. & Syst. , E78-D,**11**,(1995) 1362–1368
7. <http://www.delmia.com/>
8. <http://www.h-anim.org/>
9. <http://www.motionanalysis.com/>
10. T. Yabe, K. Tanaka: Similarity Retrieval of Human Motion As Multi-stream Time Series Data. International Symposium on Database Applications in Non-Traditional Environments (DANTE'99) (1999)

Human-Body Motion Simulation Using Bone-Based Human Model and Construction of Motion Database

Hiroshi Arisawa¹, Takako Sato², and Takashi Tomii¹

¹ Graduate School of Environment and Information Sciences,
Yokohama National University,

79-7, Tokiwadai, Hodogaya-ku, Kanagawa, Japan

² Graduate School of Engineering, Yokohama National University,
79-5, Tokiwadai, Hodogaya-ku, Kanagawa, Japan

Abstract. This paper presents motion simulation/evaluation system for factory workers in the framework of “Info-Ergonomics.” One of the key technologies is CG simulation based on the precise human body mockup called “Bone-Based Human Model.” Using BBHM, “real” motions of workers can be mapped for precise simulation. Another important issue is data and knowledge integration. For the purpose of schematizing such data and providing retrieval functions are discuss in an extended database system, “Real World Database.”

1 Introduction

Motion Simulation of factory workers is one of the core technologies to achieve optimum Human-Machine Co-existing Environment in modern factory design. IMS - HUMACS Project has been pursuing such core technologies and their principles especially from the viewpoint of “Human” aspects[1] for about last 10 years. Among them the authors group proposed “Info-Ergonomics” as an integration of simulation/evaluation technologies of human workers’ motion, in conjunction with database technologies which provides storing/retrieving functions for those knowledge[2]. We think that human centered factors are often overlooked or underestimated although human labor has an important role in the manufacturing process design. It is usually examined from the viewpoint of Industrial Engineering, Ergonomics, etc. and results are used in product planning, working analysis, and work environment design. Info-Ergonomics is a promising solution to provide total environment of work simulation and evaluation. It consists of a number of component technologies like Human body modeling, Customizing human body model to specific workers, Motion description, Capturing motions from video images, Simulation, Storing time-dependent data into databases, and Spatio-temporal retrieval on databases. This paper outlines the Info-Ergonomics concept and technologies, and explains RWDB(Real World Database) as the back-ground technology. Then we will focus on the “precise” human body modeling, called Bone-Based Human Modeling. Total system architecture to realize Info-Ergonomics will be also discussed.

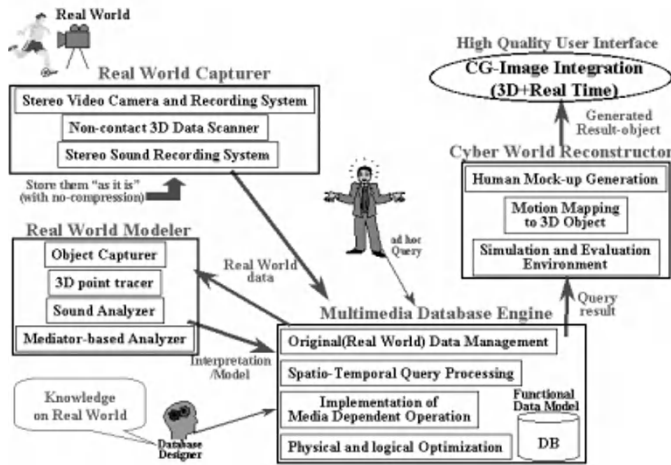


Fig. 1. Conceptual architecture of RWDB

2 Real World Database

To discuss the human body modeling and motion description, we must consider on the technical and the theoretical background of such data processing. For this purpose we proposed “Real World Database (RWDB)” as the Data processing platform. The objective of the RWDB is to provide a total environment for capturing, analyzing, storing, and retrieving physical or logical objects in the real world. Everything in the real world could be modeled through it and any type of data could be accumulated. For this purpose, RWDB must involve at least 4 components listed below. Conceptual architecture of RWDB is shown in Figure.1.

– Real World Capturer (RWC)

The objective of RWC is to capture the external form of objects in the real world. There exists various types of 3D or spatial information depending on capturing devices. The simplest one is a set of video cameras. We can get a sequence of frames from, for example, a left-eye camera and a right one simultaneously. Another type of input device is “3D Scanner” by which we can get a perfect surface (polygon) model for static objects. The practical solution is to get above two kind of informations from the real world and to combine two models into one in the database level.

– Real World Modeler (RWM)

RWM is a set of tools each of which analyzes original frame images and generate a new information. For example, the Outline Chaser[3] catches the outline of an object in a certain video frame, and then trace the outline in preceding and successive frames. Many algorithms are investigated and evaluated for range image generation in Image Processing area [4]. All the results of analysis are stored into database preserving the correspondences to the original images.

- Multimedia Database (MMDB)

MMDB is a database which treats a variety of data types such as full texts, graphic drawing, bitmap images and image sequences. The features of such data are quite different from the conventional DBMS's ones, because some of them are continuous and might occupy much more space than traditional data types. As to data model, in order to integrate all types of data, introduction of simple primitives to describe real world entities and the associations between them are essential. Moreover, the query language of multimedia data handling must involve various types of media-dependent operations for retrieving and displaying. Especially, in RWDB, the result of a query creates a new "Cyberspace". A query may retrieve a certain unit work and project it to another human worker. The author proposed a total data model to describe the 2D or 3D data, and also presented query language MMQL for flexible retrieval [4].

- Cyber World Reconstructor (CWR)

The result of database consists of various types of data such as frame sequence and 3D graphics data. In order to visualize the query result, RWDB should provide a "player" of result world. CWR is, in this sense, an integrated system of 3D computer graphics and 3D video presentation. Unfortunately, the modeling method of objects in the field of 3D graphics and VR systems are quite different from the DB approach because the former is focusing on natural and smooth presentation of surfaces of objects and their motions, whereas the latter makes deep considerations on semantic aspects of these objects.

3 Info-Ergonomics Modeling

Based on the above RWDB Concept, we concentrate ourselves into modeling on human bodies and motions especially for factory workers. We offered Info-Ergonomics for storing Human Working Data and reconstructing it in CG Simulation.

3.1 Info-Ergonomics

Info-Ergonomics is a framework of information integration for conceptual design of Manufacturing Systems. Recently modeling the machines in the factory, creating virtual machines by using CG, simulating their work and evaluating it are coming into use gradually [5]. But modeling human beings and creating "virtual employee" by CG still has very limited use, because of the human body's high complexity and the limits imposed by the computer techniques. We focus on human body modeling and cooperative works with machines, especially from the viewpoint of comfortability, safety and efficiency.

Info-Ergonomics provides the designers with modeling, evaluating, and visualizing platform to design the optimized working environment for the factory employees. From the viewpoint of IMS-HUMACS Project, Info-Ergonomics is considered as a liaison between traditional simulation/evaluation technique and highly mental aspects. (See Figure 2.)

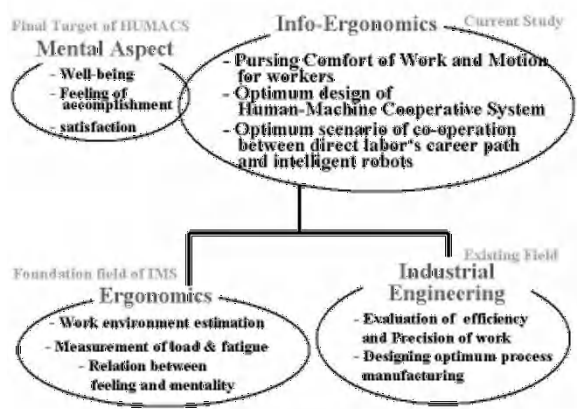


Fig. 2. Info-Ergonomics for Factory Workers

As the whole informations are schematized and accumulated based on the RWDB concept, the user can extract arbitral part of them and reconstruct 3D objects and simulation data adapting to requirements of application.

The use of CG has significant advantages. For instance, Info-Ergonomics-based simulations allow more precise evaluation than real measurement because virtual employees can be made to do any more (even those causing pain, dangerous ones, etc) and, in addition, we can perform the evaluation itself at a much more detailed level visually.

Replaying the skilled employees' work from Database to CG enables us to study body-motion from all angles intuitively. In addition, we can keep safety and work space for employees, and design comfortable work environment.

3.2 The Human Body and Motion Modeling

In order to store and retrieve Human motion data, we need an integrated datamodel which can describe all kind of objects and materials about the human body, for use in all applications. Creating realistic model of the human body requires considerable amount of data because of object's complexity. Therefore from ergonomical point of view we must re-modeling the human body with small number of primitives. We call it "Simplified Human Body", which involves, for example, simplified head, arms, body and legs, connected by small number of joints each other. The human body and motions can be described as follows.

- 1. Surface Model
Human body's figure can be described by a set of Polygon and Texture Models of body-parts, that is, each part of human body can be defined as CG surface model. An example of polygon model and texture image are shown in Figure.3(a)(b).
- 2. Structure Model



Fig. 3. Human Model

Human body consists of more than 200 bones, which are connected so that the body can move using the power of the muscles [6]. But mechanism is too complex to model it completely. We focus on movable joints and select 24 of them in order to create a simplified model of the human body, Simplified Human Model (SHM). Also we can define much more precise human body model (called BBHM) as discussing later. As for SHM, it can be regarded as hierarchical parts system, that is, each child component (usually a bone) of SHM has its own coordinate system, the origin of which means the connecting point (i.e. joint) to the parent component. An example of SHM connections between arm parts are shown in Figure.3(c).

3. Motion Model

For a certain employee's work, the action of each component of human body should be traced and modeled under the restriction of inter-joint structure. This Dynamic Model is defined on the Simplified Human Model discussed above. Each joint moves along with time. That is, the motion model of each workers (action) can be defined as the time sequence of joint-angles. On the other hand, it seems to be very difficult or impossible to define BBHM movement directly because of its complexity. Converting mechanisms from SHM motion to BBHM one will be discussed in the section 4

4 Simulation with Precise Human Mockup

4.1 Objectives of Precise Human Modeling

As discussed in the section 3, the objective of info-ergonomics is to explore comfort actions, etc. for the purpose of providing a platform that can be applied to all fields where human actions are handled (such as medical service, manufacturing, arts, and sports). Application of info-ergonomics to precise analysis of human beings at work sites such as factories will require a doll ("mockup") that not only has skin and hair, which is externally visible, but also has the internal structure of human body, especially bones, muscles, tendons, and others (flesh, blood vessels, nerves, etc.), together with freedom of joints. This research has therefore proposed and worked out a framework for

the shape and structure of a detailed virtual human being that can be actually utilized on the scene of medical treatment. When medical data is handled in info-ergonomics, there is a desire to visually simulate operations and rehabilitation in advance, using CG, with respect to a large number of humans, reflecting their individual differences. However, since data on shape, structure, and actions of humans in the real world amounts to huge volumes, it is necessary to schematize (represent) such data before simulation in order to create a database of human information. In addition, since individual human beings are different in height and arm length, we need a mockup that allows interactive changes to parameters for individual parts of human body. If the structure and actions of human body can be systematically stored on computer in this manner, it will become possible to analyze lumbago of workers and record movable range of joints and other actions to check progress in treatment.

Various human mockups have been proposed to represent human actions on computer. One of these mockups represents human actions mathematically, another represents each joint as a point, still another draws a line between one joint and another, and yet another mockup expresses each segment (each part of the body) as a rigid body. However, these mockups either have rough contours as a result of reduced amounts of information or fail to reproduce essentially smooth movement of human body because of an extremely limited number of joints. What is more, since these mockups cannot represent internal structure (bones, muscles, tendons, nerves, etc.), they can hardly meet the medical need for data as to which joints are under load or how the movements of bones are. Some of the mockups certainly represent bones and muscles of human body, but since being based on the analysis of only a part of human internal structure, they are not instrumental in the understanding of the exterior (skin) of human body. Moreover these mockups do not permit changes to scales of individual parts so as to represent humans of different heights and arm lengths. As a result, a mockup that can meet our demands must be a 3D CG human mockup that has detailed contours, can faithfully reproduce the movement of skeleton, permits interactive editing of different body structures and movements of individual humans, and allows observation from all angles.

To this end, we first created a precision mockup based on human skeleton that can reflect different heights of individual humans and varied ranges of movements of their joints. By changing the angles of joints of this skeleton mockup, we simulated a series of actions of humans at work and practicing sports. Prototype of such precise human body is displayed in Fig. 4.

4.2 Bone-Based Human Model

In order to simulate and evaluate human motions, precise 3D human body model should have internal structure with the rigid bones and joints, and flexible surface structure is required. We propose Bone-Based Human Model (BBHM) as one of such models.

BBHM is a 3-dimensional object which consists of the parts representing the bone shape of a human body, and an outer skin expressed by the flexible surface model. It have 155 parts, 76 joints, and 126 degrees of freedom. Parts are connected each other by movable joint which reflect the human body's motion. Almost all human body motions can be expressed precisely except a motion of ribs at the time of a breath etc. Using such



Fig. 4. Bone-Based Human Model

model, we can achieve precise load evaluation taking physique, joints' suppleness, and so on, into account.

4.3 Modeling and Schematization of Bone-Based Human Mockup

Based on the above discussion, a schema of BBHM, expressing the main point of human body structure is shown in Fig. 5. Also a “real” bone structure corresponding to Fig. 5 is displayed in Fig. 6.

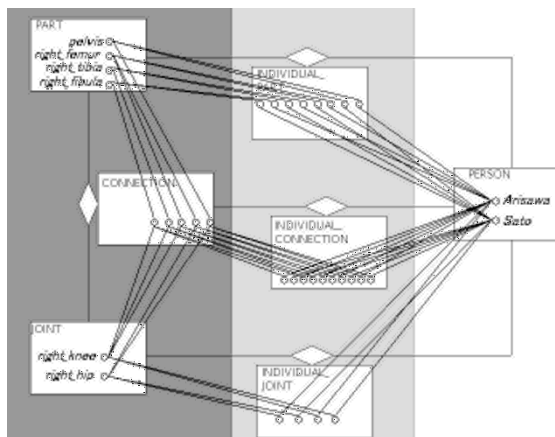


Fig. 5. expression of bone structure on database with instance examples

Bones and joints, which are independent sets of instances, are expressed with **PART** and **JOINT**, respectively. And these Cartesian aggregation, **CONNECTION**, express

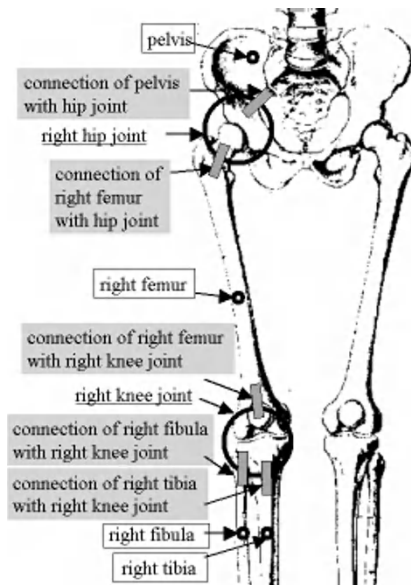


Fig. 6. bone structure of human-body

sets of connection. By relations among PART, JOINT, and CONNECTION, general human body structure can be defined.

For example, right_femur has connections between right_hip_joint or right_knee_joint. Right_knee_joint is constituted by three connections among right_femur, right_fibula, and right_tibia.

Now, we can express the person's bone with Cartesian aggregation between human body parts (PART) and PERSON, which type name is INDIVIDUAL_PART. Characteristics of bone, for example, length, are attribute of INDIVIDUAL_PART. JOINT and CONNECTION are considered similarly.

As a result, we can schematize human body structure on a bone level.

4.4 Design and Realization of a Simulation System

To achieve human motion evaluation, we must embody three types of requirements:

1. precise simulations of actual motion in a certain situation
(ex. simulation of rehabilitation clients, etc.)
2. motion mapping from a person's one to another
(ex. an amateur dancer simulates a motion of a professional dancer, etc.)
3. extracting typical motions from a specific group of samples
(ex. workspace design for "aged" workers, etc.)

Based on the concept we proposed in the preceding section, we implemented a BBHM simulation system by using engineering simulator ENVISION version 5.0 (deneb

corp., the U.S.A.[7]). The conceptual architecture and data flow of this system is shown in Fig. 7.

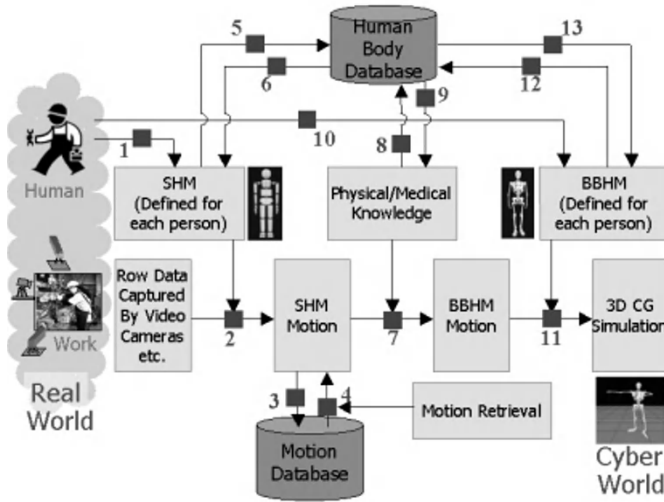


Fig. 7. Spacio-Temporal Analysis and Motion Modeling for Human Body

In this diagram, black rectangles denote “converters” or “modules”:

- 1,10 : measurement module for the human body
- 2 : motion capturer
- 3,5,8,12 : data loader (to database)
- 4,6,9,13 : database query processor
- 7 : mapper
- 11 : graphic simulator

We define human body motions on SHM firstly, then convert them to BBHM motions by using “mapper” and finally execute simulation/evaluation on the BBHM. Because motion data and mapper parameters are strictly schematized, we can various types of motion simulations on various types of human bodies.

We aim to evaluate both actual motion and imaginary motions. So we take two categories of motion definition:

1. obtain “real” motions from video images
2. define motions artificially by operating “3-D Mouse”

Every part's size, ranges of motions, and mapper parameters, are stored in the database, we can perform the simulation united with the individual. The default mapper parameters stand on medical knowledge provided by [8].

4.5 Motion Data Convert System and Motion Database

It is possible to express correctly the load state of each part of the human body by defining various motions to BBHM. But it is difficult to map the human motion to BBHM faithfully, because BBHM has a large number of degree of freedoms. In medical science, the ratio of each joint in a global motion of the human body was investigated. Therefore, we consider it is possible to map the human motion, which was evaluated, to BBHM correctly by using knowledge and technology based on medical knowledge. In our research, we propose "Mapper" algorithm as a method to map of the motion defined using SHM to BBHM by using dignity attachment parameters. The simplest method of calculating dignity attachment parameters from the ratio of the joint value(angle) of SHM and the method of calculating a joint value(angle) from a ratio of the joint values are follows.

$$P_{ij}(\%) = \frac{R_{ij}}{\sum_n R_{nj}} \times 100 \quad (1)$$

$$B_j = S_i \times P_{ij} \quad (2)$$

$$\left\{ \begin{array}{l} B_j : \text{BBHM's } j\text{th joint angle} \\ S_i : \text{SHM's } i\text{th joint angle} \\ P_{ij} : \text{dignity attachment parameter from SHM's } i\text{th joint to BBHM's } j\text{th joint} \\ R_{ij} : \text{rate at which SHM's } i\text{th joint angle effects BBHM's } j\text{th joint angle} \end{array} \right.$$

SHM Motions are stored in motion database. Motion schema is shown in Fig. 8.

4.6 Motion Simulation

As an example of motion evaluation on BBHM simulation system, we compared measured motion data with range of motion of BBHM. For example, we simulated thigh motion, which range of freedom is restricted by knee's angle (Shown in Figure 9.)

5 Conclusion

In this paper we described a motion analysis and modeling technique based on a precise human model called BBHM. We also described a motion definition technique, SHM and Mapper. As a result, detailed description about human body and various simulation of human body motion are possible with database.

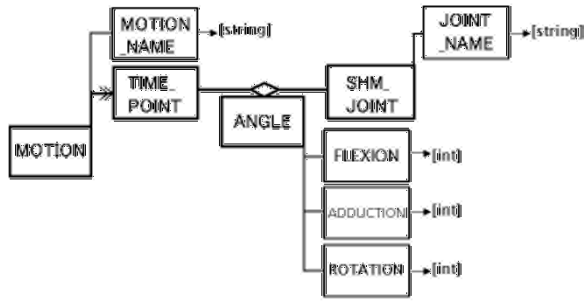


Fig. 8. Motion Schema

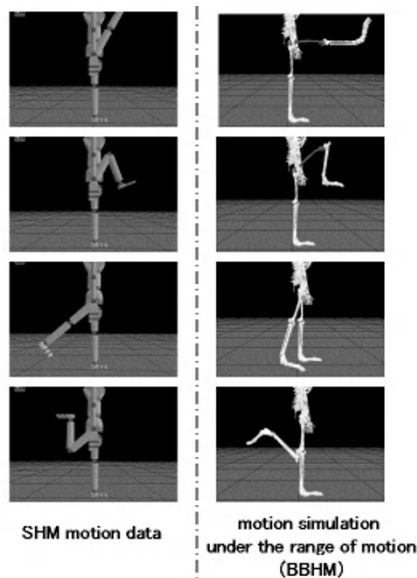


Fig. 9. evaluation of change of range of motion

Many problems are left for future. Implementation of components of human body other than skeleton model, motion analysis based on inverse kinematics and visualization of a load and a amount of fatigue about human body are remained as future problems. By modeling of the other components of the human body such as tendon and muscle, it becomes possible to evaluate human motions more precise and it becomes a lead to understand diseases and factors, which restrict degree of freedom of joints, visually. In this paper, we proposed the human body model as rigid model. Modeling of flexible surface of the human body is also open for future.

Acknowledgement. A part of this research is supported by the IMS-HUMACS project Japan. Also, this work is partly supported by a Grant-in-Aid for Scientific Research (No. 12558026).

References

1. Results of Research on Organizational Aspects of Human-Machine Coexistence Systems(HUMACS) in Manufacturing Systems (Summary Report). Manufacturing Science and Technology Center, IMS Promotion Center (2001)
2. Arisawa, H., Imai, S.: Mediator-Based Modeling of Factory Workers and Their Motions in the Framework of Info-Ergonomics Human Friendly Mechatronics (Selected Papers of ICMA2000). 395–400 (2001)
3. Michihiko Hayashi, Takashi Tomii, Hiroshi Arisawa: A Form Acquisition of Subjects in Image Databases. IPSJ SIG Notes Vol.97, No.7, 97-DBS-111-13 (1997)
4. H. Arisawa, T. Tomii, H. Yui, H. Ishikawa: Data Model and Architecture of Multimedia Database for Engineering Applications. IEICE Trans. Inf. & Syst. , Vol.E78-D No.11 (November, 1995)
5. Kageyuu Noro: Illustrated Ergonomics. JIS (1990)
6. R. Nakamura, H. Saitoh: Foundmental Kinematics. Ishiyakushuppan Cop. (1995)
7. <http://www.delmia.com/solutions/html/ergonomics.htm>
8. I. A. KAPANDJI: Physiologie Articulaire. Maloine S. A. Editeur (1980)

Ontological Commitment for Participative Simulation

Jan Goossenaerts and Christine Pelletier

Eindhoven University of Technology, Faculty of Technology Management, POBox 513,
5600 MB Eindhoven, The Netherlands
{J.B.M.Goossenaerts,C.P.M.Pelletier}@tm.tue.nl

Abstract. This paper analyses the role of ontological commitment in structuring the requirements for the PSIM Environment. This environment aims to support (i) the sharing and the exchange of knowledge between the different actors involved in the design or redesign of a manufacturing enterprise; and (ii) the exchange of information between tools supporting enterprise analysis according to different perspectives (logistic, technologic and human). The techniques for piecemeal ontological commitment are related to two contributions from research on enterprise reference architectures: (i) the dimension of genericity of the ENV 40003 reference architecture and (ii) the relationships between lifecycles of enterprise entities as defined in GERAM. Two kinds of applications illustrate the ontological commitments: support for the interoperation and communication between applications; and the provision of task-specific interfaces to users working in an enterprise.

1 Introduction

This paper presents some results of the European component PSIM of the IMS/RTD project “Organizational Aspects of Human-Machine Coexisting Systems” (HUMACS). PSIM is developing and pilot-demonstrating a Participative Simulation environment for Integral Manufacturing enterprise renewal. Integral means that the analysis involves logistics, technology and human factors. PSIM focuses on assisting human creativity and the application of ICT within assembly organizations: to enable the knowledge processing to make the work methods of a manufacturing enterprise reflect *all manufacturing expertise* that is available in the organization [1]. As the organization’s processes have increasingly been supported by software tools and applications, the four types of knowledge conversion of the SECI model: socialization, externalization, combination, and internalization [9] will also involve these software’s data and processing. The enterprise’s IT environment including the simulation tools are often considered assets as key carriers of explicit knowledge. However, due to poor support for people with various experiences to participate in the analysis, and due to poor connectivity to enacted models and real life data, the IT environment is often seen as a hurdle too. Hence the goal of PSIM to (re-)enhance organizational knowledge creation by accomplishing that several tools and several content bases can be combined with the transparency of “a single multi-user tool”.

The contents of this paper is organized as follows. Section 2 presents some general insights on participative simulation and the role of enterprise ontologies for

knowledge creation in organizations. Section 3 presents the principles behind the ontological commitments of the PSIM environment prototype. Section 4 presents the ontology library and section 5 illustrates the use of the PSIM enterprise ontologies.

2 Ontological Commitment and Knowledge-Creation

The PSIM approach to knowledge management is based on the role of an organization in allocating knowledge: "The organization supports creative individuals or provides contexts for them to create knowledge. Organizational knowledge creation, therefore, should be understood as a process that 'organizationally' amplifies the knowledge created by individuals and crystallizes it as a part of the knowledge network of the organization" [10:59]. According to Nonaka's SECI model [9] knowledge creation is a continuous and dynamic interaction, spiraling circularity, between tacit and explicit knowledge. This interaction is shaped by shifts between different modes of knowledge conversion. It starts from "Socialization", when workers are interacting with each other, as well as experience. Then during "Externalisation" or articulation, - the knowledge is made explicit, for example in drawings, models, their evaluations, etc. This is followed by the "Combination" step during which explicit knowledge is converted into more complex sets of explicit knowledge, for example in plans, reports, work instructions. The "Internalization" converts the organization's explicit knowledge into the organization's tacit knowledge. Then the process starts again with socialization and so on. Thus the knowledge creation happens *continuously* and in multiple concurrently operating teams. Fig. 1 represents both the primary process of the team (Arrow A) and the self-transformation (B) in which the organization abandons obsolete knowledge and learns to create new things, improves its activities and develops new applications. The B arrow applies the SECI modes of knowledge conversion.

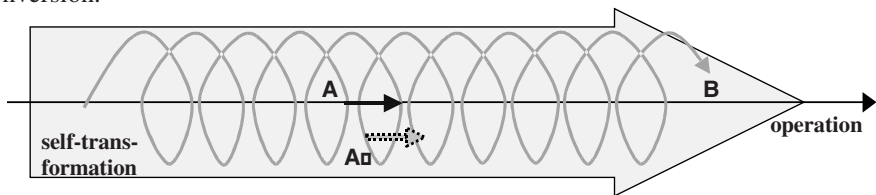


Fig. 1. Operation (A) and Self-transformation (B) in a team

The goal of *participative simulation* is to enable workers to exert direct influence over the product and process designs by bringing in their tacit knowledge, through making explicit relevant parts of it such that it can be combined with other explicit knowledge, and to put the blend of knowledge to the test, and eventually into practice. Enterprise ontologies and techniques for piecemeal ontological commitment are key to structuring various stakeholders' requirements prior to the development of a participative simulation environment.

2.1 Ontological Commitments

An *ontology* corresponds in practice to a set of formal terms, usually with a hierarchical organization, and to a set of constraints about their use in the knowledge representation of the domain studied. The formal terms are associated with formal definitions that specify their relationships with the other formal terms. Each term can be seen as a knowledge category that can be instantiated. General guidelines have been stated to help the ontology builders in their task [13]. An ontology has to be rooted in a broad mutual understanding and agreement of the different domain stakeholders. For an “abstract ontology” the stakeholders have to agree on the general hypotheses made on the knowledge domain. These hypotheses deal with the different high level categories or concepts, which are used to express the objects or object types and their relations [8]. A stakeholder, or agent, commits to an ontology if its observable actions and objects are consistent with the definitions in the ontology. In the case that the stakeholders have different purposes when deploying the ontology, their ontological commitment may vary. A possible state of objects and possible actions or state changes are often referred to as a *possible world*.

The *ontological commitment* of an agent or stakeholder can be defined as the ontology that allows the agent to distinguish the possible objects, actions and state changes, as well as the rules governing the dynamics, in his or her possible contexts of work, operation, or reasoning.

Depending on the kind of stakeholder, this definition can be made more specific. Ontological commitment is defined as a formalized mapping between terms in a knowledge-base and identical or equivalent terms in an ontology in the context of knowledge-based systems that –in order to interoperate– must commit to the definitions in a common ontology [17].

In the context of the “PSIM-enabled-knowledge-creating company” at least the following four kinds of stakeholders are identified.

The *employees* form one group of stakeholders, each with their private view of the enterprise, or *Private Ontological Commitment (POC)*. At least during working hours the POC must be partly aligned with the ontological commitment of the enterprise, for instance in order to enhance knowledge sharing and contribute to the value adding processes of the enterprise.

The *enterprise* itself, viewed as a juridical, financial, technical and social entity producing goods, is considered a second stakeholder. In one extreme case it could be represented by the decision maker, committing the enterprise to a new or revised common ontology –which better fits the business reality–. The common ontology is called the *Enterprise Ontological Commitment (EOC)*. An employee’s POC will be partially derived from the EOC, which is shared among the employees. The EOC can be more or less elaborated. When they are present, ERP systems, MES systems, workflow management systems and other enterprise applications will also embody projections of the EOC.

Knowledge domain engineers form a third group of stakeholders. For instance the ergonomics research discipline defines suitable loads for physical work. Ergonomic knowledge is applicable in all situations where people work. The term *Domain Ontological Commitment (DOC)* is used in the description of the commitment of this domain knowledge to the ontology. Accounting and logistics are other domains with

their own DOC. To the extent that tools embody expertise of a knowledge domain, their ontological commitment can be mapped to that of the domain.

The *PSIM environment engineer* is a fourth (group) of stakeholder(s), the ontological commitment of which is the focus of this paper. In this case the term “*Participative Simulation Ontological Commitment*” of PSOC is used.

2.2 Applying Piecemeal Ontological Commitment

Minimal ontological commitment has been advocated as an ontology design principle but Borst et al. [2] have pointed at the risks of “over-commitment” and “under-commitment”. “Over-commitment” means that the set of possible worlds allowed by the ontology is too restrictive. So it leads to reduction of reuse and sharing. “Under-commitment” means that the set of possible worlds may include undesirable worlds, causing poor end-user guidance and support. The balancing act between under- and over-commitment requires: “*piecemeal ontological commitment*”. Starting from the minimal ontology, one needs to incrementally build up the ontological commitments until the right degree of commitment is achieved. Several techniques to achieve piecemeal ontological commitment have been proposed [2]: (i) The use and reuse of “super” theories. These theories are general and abstract ontologies such as mereology, topology and general systems theory; (ii) The identification of various viewpoints or base ontologies. They are broad and stable conceptual distinctions and are seen as natural within a large domain and give rise to a categorization of concepts and properties; and (iii) The use of ontology projections as a means to formalize the interdependencies between included ontologies.

In an environment for participative simulation, techniques for dealing with ontological commitment need to address the typical challenges for each kind of stakeholder, and the typical mappings or projections between the ontological commitments of different stakeholders.

The POC of *employees* and the EOC of enterprises is confronted with the following challenges: (i) Ontological heterogeneity resulting from the varying competencies of the workers and tools they use; and (ii) The growing importance and increasing frequency of changes to the enterprise operations, one cause of such changes are improvement procedures.

Knowledge domain engineers have to cope with (scientific) progress within the domain, which may lead to incremental or radical changes in the DOC.

The *PSIM environment engineer* has to come up with solutions for the following problems: (i) The existence of many relevant knowledge domains for enterprises, the product and process related domain and also domains such as socio-technical systems design and ergonomics and the reduction of physical and mental workload; and (ii) The enormous diversity of the particular enterprise domains, especially as more knowledge intensive tasks are considered.

Projections exist between the EOC and work-related part of the POC of its employees. Domain ontologies, e.g. DOC1 and DOC2, must be linked to the EOC of an enterprise, prior to their application within the enterprise. The “low-commitment” generic ontology of the PSOC makes it possible also to establish partial mappings between DOC1 and DOC2. Viewpoints or base ontologies matter for the Domain

knowledge engineers and for the PSIM environment engineer. Indeed, the PSIM environment has been committed to enterprise applications and their stable conceptual distinctions. The following three sections will further illustrate piecemeal ontological commitment and ontology projections for the different stakeholders.

3 Ontological Commitments for PSIM

This section answers questions such as “What is the enterprise ontology with a minimal ontological commitment?” and “What are suitable increments in the ontological commitment, such that any enterprise, worker or tool can be supported in establishing interfaces with the “organizational knowledge”?”

In PSIM we have first focused on *partial ontologies of individuals or particulars* with no attention for classification of the objects and relations between them. The initial choice for ontologies of particulars was based on three considerations: (i) The reduction of complexity while emphasizing opportunities for communication between engineers, managers and employees; (ii) The preference of addressing all dimensions of a more simple problem solving-scenario— all the steps in the SECI model – over offer improved support for a single step in such a scenario; and (iii) The need for a sound foundation for enterprise ontologies, in terms of its primitive elements and constructors. Guarino [7] has adopted a similar approach of addressing particulars first in investigating identity conditions in ontology design.

For PSOC, our initial focus has been on the following four principles: Heraclitus’ principle of flow and GERAM’s principle of Entity life-cycle; Plato’s principle of continuant and occurrent; Plato’s Form and CIMOSA’s Generic, Partial and Particular Models; and Peirce’s Principle of Semiotics. These four principles allow the PSIM environment to support simple participative simulation scenarios with ontology sharing for EOC, POC and DOC, but without using classification, generalisation (principles that we owe to Aristotle) and aggregation in the ontology sharing. The PSIM environment itself makes use of the latter principles, and so may the other stakeholders for their “internal work”.

3.1 Heraclitus’ Flow and GERAM’s Enterprise Entity Life-Cycle

“Everything flows and nothing abides” is a key thought in Heraclitus’ (approx. 540-475 BC) perception of the world. Moreover, the change is governed by logos, or reason. Applying several more recent ontological commitments, a more refined holistic perspective is visible in the definition of the life-cycle of enterprise entities of GERAM [5]. The GERAM life phases of enterprise entities are reflected in two partial PSIM ontologies. One partial ontology defines concepts for the *operative processes*, these are all enterprise processes during the *Operation phase* of their life cycle. For example, the design process, applied to new products and/or manufacturing systems, is an operative process (see also the A arrow in Fig. 1). Another partial ontology, the PSIM-procedure model, defines concepts for the *self-transformation processes*. These are all enterprise processes during the *Identification, Concept, Requirement, Design, Implementation and Decommission phases* of their life cycle.

Therefore the redesign and/or optimisation of a manufacturing process are a part of a self-transformation process (improvement or renewal process) (the B arrow in Fig. 1).

In current IT environments, the concepts for operative processes are to a large extent covered by ERP systems. The additional purpose of the PSIM-procedure model is to allow the enactment of systematic practices for the management of self-transformation within an organization. As this partial ontology is further developed it will allow the enterprise to enact its particular variant of the SECI model, and to benefit from IT support for it. The term *ontogenic development* is used to refer to both the operations and the self-transformation of an enterprise entity.

3.2 Plato's Principle of Continuant and Occurrent

The classical distinction between continuants and occurrents [7] can be traced back to syntactic categories in natural language, appearing as early as in Plato's (427-347 BC) *Sophist* Dialogue. Meaningful statements or propositions (*logos*) are obtained by joining on the one hand a token that is called the *verb* (or *predicate*) and represents an action, and on the other hand there is a token that is called the *noun* and represents the agent that performs, or the object that undergoes the action.

In enterprise integration and open systems architecture [3,4,18], this principle is present in the resource view and function view [3,4]. This work, and their viewpoint ontologies have been applied in tools supporting ERP implementations [12,14].

In the PSIM enterprise ontology kernel, the principle is reflected in the two generic interrelated entity classes Activity and Object. *Activity* captures the notion of anything that performs something by transforming its *inputs (objects)* into *outputs (objects)*. This transformation may happen only if some *conditions* are verified [3,15]. *Object* is something, which allows the realization of some Activities, when it is available. The main feature of this entity is not being available at anytime [15].

The relationship that can exist between these two generic entities is called involvement. *Involvement* is a general relationship. It represents the fact that an object can be involved in the application of an activity. This involvement can be further specialized as an input, output or resource involvement, for example.

Because we considered available organizational and distribution concepts insufficiently integrated with those of activities, objects and relationships, we have made the simplification to consider an organizational unit as information about the organization of objects. This abstraction is acceptable given the current focus and purpose of the PSIM project. The introduction of the concept of space is currently investigated, as an additional piece of the PSOC.

3.3 Plato's Form and CIMOSA's Generic, Partial, and Particular Models

Plato held that necessary connection supporting valid inference for statements depends on forms, which can roughly be described as a character or set of characters common to a number of things. In enterprise modeling, commonality as a basis of valid inference has been one of the origins of the dimension of genericity in enterprise integration and open systems architectures [3,4]. In PSIM a partial ontology is

constructed, for instance the one in Fig. 3, as a refinement of the generic ontology in Fig. 2. This construction requires that specialization trees are defined for all the entities and relationships in the generic ontology. Inference rules that are valid for a more generic ontology are also valid for the ontologies constructed from it.

3.4 Peirce's Principle of Semiotics

While the principles of continuant, occurrent, and form enable us to capture knowledge, they do not provide a clue of how to form and apply such knowledge. The latter problem is addressed by Peirce who states that semiosis – the process in which knowledge is formed and applied – involves three factors: that which acts as a sign (the sign vehicle), that which the sign refers to (the designatum, the signified or object), and that effect on some interpreter in virtue of which the thing in question is a sign to that interpreter (the interpretant) [11]. For more details on semiosis and its relevance for industry and IT deployment, see [6]. Applying the principle of semiotics to the typical setting of participative simulation involving employees in an enterprise context, the following mapping is easily made. The object(s) are the (future) “enterprise entities in operation”. The signs are both the instances of *Object* and *Activity* and the information elements related to these instances; the instances refer to the enterprise entities in operation. And the interpreters are the employee-users and software applications in the enterprise. The employee-users map signs to the enterprise entities and the operations.

The mapping involves two domains, which could be characterized as physical and informational. Within the information domain the generic entity class Information Element is added. An *Information element* is a characteristic of anything, e.g. an object or activity or another information element, which is used to express directly or indirectly knowledge on their mutual involvement, e.g., of the objects, activities and other information elements.

Between information elements on the one hand, and their referents on the other hand, there exists a *Relevance* relationship expressing that an information element is related to an object or an activity, or to another information element. For example the weight of a piece of material or the time needed to run an activity. Information elements can be used to express the conditions for an object instance to be involved in an instance of activity.

4 The Ontology Library

First, we describe a generic ontology that combines the ontological commitments of semiotics and of continuant and occurrence. Next we focus on three partial ontologies that together cover the enterprise life-cycle and the domain ontology for the ergonomic analysis of a task. Together these partial ontologies serve as a starting point for constructing the particular ontology of an enterprise.

4.1 The Kernel Proper or a Generic Ontology

A manufacturing enterprise, as any enterprise, can be described by means of instances of the three generic interrelated entity classes: Activity, Object, and Information Element (see also Sections 3.2 and 3.4). Fig. 2 shows the schema of the concepts. The relevance relationship has been split into three relationships depending the elements it links: I-O-, I-A- and I-I-relevance.

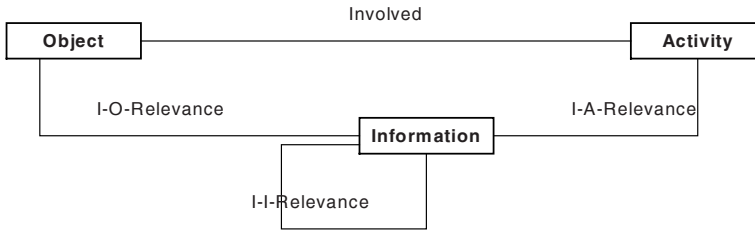


Fig. 2. General ontology schema

The partial ontologies in the PSIM Ontology Library are positioned at the partial model layer in the ENV 40003 dimension of genericity [4]. An important requirement for a PSIM partial ontology is that *it must specialize all the generic entities (object, activity and information element) and all the generic relationships in accordance with its competence area*.

Two partial ontologies cover the entire life cycle of the enterprise: the PSIM-ontology for enterprise operations and the PSIM-procedure model.

4.2 Enterprise Operations

In the Operations partial ontology, the CIMOSA enterprise modeling entities and relationships [3] are defined as specializations of the entities and relationships in the generic ontology kernel. The more specific entities include resource, product, routing activity, capability; the relationships include input, output, is responsible of. The ontology's competencies are related to the support for primary process operations, the description in more detail of the enterprise and its principal features on the level of the primary process operation. Fig. 3 includes the richer CIMOSA enterprise modeling entities and relationships.

A *regular activity* has objects as *input* and *output*. It has a *procedure* that shows the sequence of *steps* a *resource* has to follow and the tools (*technical resources*) s/he/it has to use, when this resource performs the regular activity. The *resource* can perform a *regular activity* if and only if its *profile* includes the *capabilities* needed by the *regular activity*. The *procedure* of a regular activity refers to *steps*. The latter concept is defined in the activity taxonomy. A *logic activity* has also a *procedure*. This procedure describes the conditions linked to the routing of the output of the preceding regular activity into the regular activity (activities), which follows.

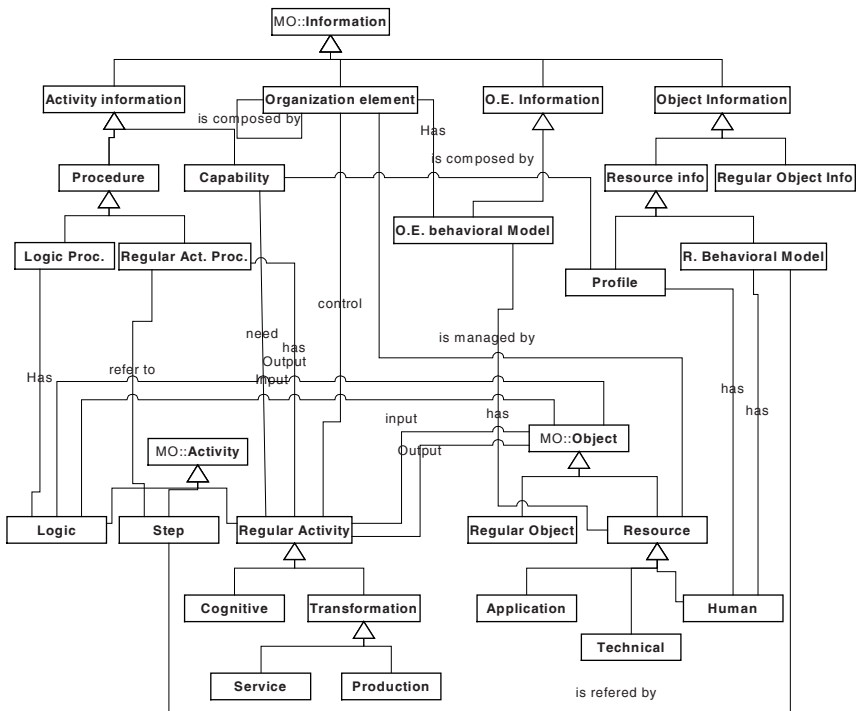


Fig. 3. Detailed ontology

The *organization of the enterprise* is described via a set of linked *organization elements*. Each organization element can be responsible of other *organization elements*, manages *resources* and controls *activities*. It has a *behavioral model* that explains the way in which it manages the *resources* and controls the *activities*.

4.3 Improvement and Renewal Procedure

A second partial ontology addresses modeling in support of change procedures: the PSIM procedure. This partial ontology should leverage the scientific basis on which to model social and human factors as an integral part of the renewal of manufacturing systems. A PSIM procedure contains the following stages: Selection of a topic or problem, Selection of participants, Ontological convergence, Definition of goals, Idea generation, Simulation of the ideas, Evaluation, and Selection of solution. Each of these stages corresponds to a cognitive activity in the ontology described in Fig. 3. These activities are all followed by service activities, which are provided by the system (management of data, rights...). A user is modeled as a resource, and a profile describes his rights in the PSIM environment. The tools embedded in the environment are also resources. The regular objects used as input and output are data.

The PSIM Environment has to support the problem-solving process by informing the participants about meetings, coordinating possible dates, sending reminders about work packages which have to be done *etc.* This moderation is necessary to keep the improvement process going on, even when the workload endangers its progress.

4.4 Task Analysis in Ergonomy

The specific aspects of fitting tasks and technology to human operators is dealt with by an ergonomic approach concerned with optimizing the tasks, technical systems and work stations in order to improve human performance and to reduce mental and physical workload. Some important aspects in the reduction of workload are the good fit between task and personality, possibilities to develop and regulate your own work [16]. Ergonomic evaluation tools must warn users when unacceptable workload for humans and teams is anticipated in a particular work system design.

The presence of the concept *step* in the taxonomy *activity* (Fig. 3) is due to the level of detail needed to perform these analyses. The concept *step* appears in sequence in the description of the procedure of an activity, and never as an isolated element in an ERP system for exampl. To support the exchange of information between two perspectives, the relationship between the concepts used is included in the ontology.

5 Deploying the Ontology Library

The ontology projection corresponds to a linkage between a core ontology and a specialized ontology. The Enterprise ontology (EOC) corresponds to a description of the enterprise within the logic and the structure of the core ontology (PSOC and relevant DOCs). The projection of the ontology for each other domain is obtained by the use of filters.

The filters used to connect tools using different paradigm to analyze the enterprise entities are called “translators”. These translators embody the relationships between the local world in which a specific tool can be used, and the ontology shown in Fig. 3. To link the POC of the employee or the DOC of the expert to the enterprise ontology, the same kind of filter is used.

The piecemeal ontological commitment enables conceptual modeling within enterprises, such that models and data of the enterprise can be seamlessly deployed in different applications, by different end-users for a wide range of tasks. In this section our focus is on illustrating the use of the Ontology Library. Three issues are addressed: (i) the complementarity of the operations and improvement ontologies in the EOC; (ii) knowledge domains (DOCs), tools and translation support for them; and (iii) EOC-based navigation support for end-users.

5.1 A Simple Enterprise Ontology with Operations and Improvements

We present an example of a very simple enterprise transporting bricks from a stockpile at one side of a road to a wharf on the other side. The persons in this enterprise execute four different activities: *picks*, *drops*, *passes* and *prepares*. The

bricks are all the same and have a weight of 2kg. The enterprise operations have the aim to transport all bricks from a pile at one side of the road to the wharf at the other side. This is described by the information element location related to each brick being wharf. Initially this information element has the value pile. The particular ontology for the enterprise includes the following instances: Objects *p1*,..., *p5*, *brick1*,..., *brick1000* and *road*; and Activities: *picks*, *drops*, *passes*, and *prepares*. The activity *prepares* consists of building the human chain on the road and assigning to each person the activities s/he has to perform. *Prepares* involves in our case the five *persons* and the *road*. The activity *picks* involves as objects one *brick* and one *person*. The activity *drops* involves one *brick* and one *person*. The activity *passes* involves two *persons* and one *brick*.

Each activity, following the ontology model, has a *procedure*. The *prepares* activity i.e. the building of the chain and the assignment of activities follows this procedure: distribute the persons on the road with a proper distance in between and assign activities to the persons in function of their places in the human chain. Each activity different of *prepares* is described by only one step. For each of these steps, a *behavioral model* is defined. Often the *behavioral model* is conform to the ergonomic standard of security. It shows the different movements the person has to perform to realize the step.

To increase the speed of the transport of the bricks, the process can be changed. The process could also be transformed to increase the robustness of the operations in regard of the number of persons. These changes can modify the definition of the procedure of each activity and also the *behavioral model* referring to each of the steps. These changes must be identified and evaluated within domain views and within the integrated view of the enterprise.

The description of the simple enterprise corresponds to the enterprise ontology commitment (EOC). It is the presentation of the whole business process of the company. For each employee in this enterprise, its private ontology commitment will correspond to the view s/he has of this business process, and can be reduced for a worker to the name of the activities s/he is involved in and the behavioral model associated to each of them.

5.2 Translation Layer with Ergonomic Concepts

Another applications concerns the exchange of data between tools supporting analysis of manufacturing enterprise from different viewpoints. In each viewpoint the used language is slightly different. The translation between these languages and the common one specified by the ontology concerns the shared concepts. Because of the difference of the approach, also the terms used to design shared concepts differ. This difference can also occur on the level of details considered in the analysis or of the perspective adopted to consider the enterprise elements. In this case, the translation is more complex as illustrated below.

The simplest case we have encountered is when there exists a one-to-one relationship between terms used in different languages. That means that for a term in a language A, there is a single corresponding term in the common language, which is translated in a single term in the language B. In this case the concepts are shared 1:1. In Table 1, terms from the ergonomic paradigm, i.e., Task, Workstation, and Task Included, are translated into the common language.

Table 1. The simple translation of some terms from the ergonomic paradigm

Task	Regular activity	
	Description	Procedure the regular activity has.
Workstation	Organization element	
	Description	Behavioural Model the organization element workstation has.
	Task included	The set of activities, which are controlled by the organization element workstation.

Not all concepts do 1:1 correspond to shared concepts. Depending on the concepts, other kinds of translation are possible in the ontology we defined. For example, the difference of viewpoints can lead to a difference of granularity in the concepts manipulated in the analysis. This produces a 1:n relationship.

We were confronted with this problem when translating the term *action* used in ergonomic analysis into a corresponding term in an ERP system. The ergonomic term *action* corresponds to the notion *step* in the PSIM ontology. This notion *step* exists in the ERP system, but does not exist independently of the concept *activity*: it is used in *sequence* to describe the *procedure* of an *activity*. We enable the exchange of data concerning *action* by adding in the PSIM ontology explicitly the relationship existing between the ergonomic concept *action* and the ERP concept *step* stored in the activity procedure. This relationship is the relationship *refer to joining regular activity procedure* and *step* (Fig. 3).

5.3 Ontological-Commitment-Based Navigation

The PSIM Navigator offers the employees guidance in accordance with the relationship between the EOC and the employee's POC. After succeeding the login to the PSIM Environment, a user gets an interface that depends on his user profile (rights). From this interface, he can consult what he is allowed and/or required to perform at that moment according with his job description, and how. During the connection, the user can access the help on line concerning the activities he is allowed to perform, as described in the procedure of the activities. In principle, for each worker, this job description may include tasks of the "primary process" as well as the "improvement or renewal process". The worker will, then, choose which *task* he wants to perform, and the PSIM Environment will select the procedure related to the chosen task. According to the procedure, the user is supported in operating and provided with the right tools, in the appropriate sequence and with user interfaces selected for the specific user, according with his behavioural model. The Navigator will feed each tool with the information related to the specific step to be performed by that user at that point in time. All specialized data are found by the Navigator inside the enterprise ontology or provided by the ontology itself mining them inside either Enterprise Data Bases or Integration Data Base.

According with his role, the user will work in a different environment. When acting as an operator, to perform a specific action, the user will operate in the *real enterprise environment*, supported by tools loaded with *run time data* and operating on real enterprise information systems. When acting as a *decision maker* or as a *designer*, he will operate in the *virtual enterprise environment*, supported by tools loaded with *virtual data*, and operating on a virtual environment for analysis.

6 Conclusion and Future Work

Within an enterprise, the PSIM environment will facilitate interoperation and communication between applications and it will support the integration of the related knowledge acquisition, discovery and modeling efforts of the end-users within the context of their enterprise. PSIM has adopted an ontology based approach to the capturing and exchange of explicit enterprise knowledge, among people and software tools. Enterprise reference architectures and enterprise life-cycle concepts have been extended into a framework supporting piecemeal ontological commitment.

Minimal ontological commitments for the PSIM Environment have been presented. A “generic” and several “partial” ontologies have been constructed taking into account relationships to basic principles. Ontology projection has been illustrated as a technique for building a more specific ontology on the basis of more general ones.

Further extensions of the generic ontology are related to the so-called “super-theories” such as set theory and mereology and the concept of space. Such theories will have to be included into more powerful and more complete generic ontologies.

In order to consolidate the PSIM environment architecture and its support for all knowledge conversion phases of the SECI model, the focus has been on basic scenarios, with relatively simple data capture techniques. Other Humacs work packages address more advanced data modeling techniques and scenarios to which the architectural choices for the PSIM environment could and will be extended in future projects.

Acknowledgements. This work has been partly funded by the European Commission through IMS Project *PSIM* (No. IMS 1999-00004). The authors wish to acknowledge the Commission for their support as well as the contribution of HUMACS and PSIM project partners to various ideas and concepts presented in this paper.

References

1. Berg, R.J. van den, Eijnatten, F.M. van, Vink, P., and Goossenaerts, J.B.M.: Leveraging human capital in assembly organizations: The case for participative simulation. Proc. IST conference, Helsinki (1999).
2. Borst, P., Akkermans, H., and Top, J.: Engineering Ontologies. Int. J. Human-Computer Studies, 46 (1997) 365-406
3. CIMOSA, ESPRIT Consortium AMICE, editor. CIMOSA: Open System Architecture for CIM. Springer Verlag, Berlin, 2nd, rev. and ext. ed edition (1993)
4. ENV 40003. Computer integrated manufacturing -systems architecture- framework for enterprise modelling. European prestandard, CEN/CENELEC (1990)
5. IFIP-IFAC Task Force. GERAM: Generalised Enterprise Reference Architecture and Methodology. Version 1.6.1, May 1998
6. Goossenaerts, J.B.M.: Industrial semiosis – founding the deployment of the ubiquitous information infrastructure. Computers in Industry, 43 (2000) 189-201
7. Guarino, N.: The Role of Identity Conditions in Ontology Design. In C. Freska, D.M. Mark (Eds.) COSIT'99 LNCS 1661 (1999) 221-234, Springer-Verlag
8. Guarino N., Carrara M., Giaretta, P.: Formalizing Ontological Commitments. In Proc. Of AAAI'94, Seattle, Washington (1994)

9. Nonaka, I.: The Knowledge-Creating Company, Harvard Business Review Nov-Dec (1991) 96-104
10. Nonaka, I., and Takeuchi, H., *'The knowledge creating company'*, Oxford University Press, New York (1995)
11. Peirce, C.S.: Collected Papers of Charles Sanders Peirce, (six volumes). In: C. Hartshorne, P. Weiss (Eds.), The Belknap Press of Harvard University Press, Cambridge, Massachusetts (1960)
12. Scheer, A.-W.: Business Process Engineering -- Reference Models for Industrial Enterprises. Springer Verlag, Berlin (1994)
13. Uschold M., King, M., Moralee, S., and Zorgios, Y.: The enterprise ontology. The Knowledge Engineering Review 13 (1998)
14. Van Es, R.M., and Post, H.A.: Dynamic Enterprise Modelling: linking business and IT, Kluwer, Deventer, the Netherlands (1996)
15. Vernadat, F.B.: Enterprise Modeling and Integration, Principles and Applications, Chapman & Hall, London, England (1996)
16. Vink, P., and Peeters, M.: Balancing organizational, technological and human factors: the vision of production management. In: Vink P, Koningsveld EAP, Dhondt S, eds.: Human factors in Organizational Design and Management VI., Elsevier Science Ltd, London (1998) 7-11.
17. Waterson, A. and Preece, A.: Verifying ontological commitment in knowledge-based systems. Knowledge-Based Systems 12 (1999) 45-54.
18. Williams, T.J., P. Bernus, J. Brosvic, D. Chen, G. Doumeingts, L. Nemes, J.L. Nevins, B. Vallespir, J. Vlietstra, and D. Zoetekouw. Architectures for integrating manufacturing activities and enterprises. Computers in Industry 24/2-3 (1994).

Dynamic Management Architecture for Human Oriented Production System

Keiichi Shirase, Hidefumi Wakamatsu, Akira Tsumaya, and Eiji Arai

Dept. of Manufacturing Science, Graduate School of Eng., Osaka University,
2-1 Yamada-oka, Suita, Osaka 565-0871, Japan
{shirase, wakamatsu, tsumaya, arai}@mapse.eng.osaka-u.ac.jp

Abstract. In advanced factories, human workers or craftsmen are required to achieve highly flexible and skillful production. This paper introduces a dynamic management architecture for human oriented production systems. The proposed architecture includes two technologies; one is a communication network based on an active database system, and the other is a dynamic scheduling method to reflect the workers' satisfaction, ability and flexibility. It has been verified by several case studies that the dynamic management architecture proposed is effective in realizing an efficient human oriented production system, which considers the workers' satisfaction, ability and flexibility.

1. Introduction

Because of the drastic change in computers and information technology, production systems have to be changed. Some mass production systems or transfer line production systems have become old fashioned, and YATAI production systems or cell production systems have been introduced to satisfy a great variety of customers' requirements. In these environments, human workers or craftsmen have been required to achieve highly flexible and skillful production, but also it is important to offer satisfaction to the workers or craftsmen, and a new management architecture to consider the working satisfaction is required. In this study, it is assumed that workers select desired jobs according to their value and satisfaction. It means that the decision whether a worker selects an assigned job or not is unpredictable. The process in which the worker selects the desired job is similar to the decision making process of each autonomous cell organizing the distributed production systems [1, 2, 3].

Additionally, in an advanced factory the efficiency and flexibility in production has been enhanced by integrating CAD/CAM systems, automated facilities and so on. However, the control information for the facilities needs to be generated previously using the CAD/CAM systems. Therefore, a conventional flexible production system has no way to adapt for a disturbance, such as a sudden change in the production plan or trouble in the actual production processes. In order to solve this problem, a distributed production system is constructed using an active database system, which is an integrated system of a network and a database [4].

In this paper, the dynamic management architecture for a human oriented production system, which considers worker satisfaction, ability, and flexibility, is proposed. In the proposed architecture, the workers correspond to autonomous cells in a distributed production system, and every worker can select his/her jobs and change his/her working hours interactively according to his/her feeling about satisfaction and value. Furthermore, the distributed production system can deal with the newest production order generated by the dynamic scheduler, and achieve continuous production in case of disturbance such as a sudden change of production plan or trouble with facilities.

2. Architecture of a Human Oriented Production System

The proposed system architecture in this study is shown in Fig.1. An active database system, which is an integrated system of a network and a database, is used to construct a feasible architecture for the human oriented production system [4].

The core part of this system is the active database which consists of a communication controller, a cell database and a job database. The cell database stores the ability of workers, cells and AGVs that are forecasted in the present and future, and the status that is reported from workers, cells and AGVs. The job database stores the production requirements of how many and what components need to be produced, and the present status of each job or operation.

A dynamic scheduling system and CAD/CAM systems are connected to the core part. Human workers, autonomous cells and autonomous AGVs are connected also to the core part through GUI controllers, cell controllers, and AGV controllers, respectively. Detailed information of products is defined by the CAD/CAM system using machining or assembling features with process constraints. The ability of workers and cells is characterized by the machining or assembling features treated and their performance.

Another important mechanism is communication using triggers or messages among workers, cells and AGVs. The active database delivers messages to exchange information among workers, cells and AGVs, in order to manage distributed manufacturing when triggers occur. Four types of triggers are considered; one is a

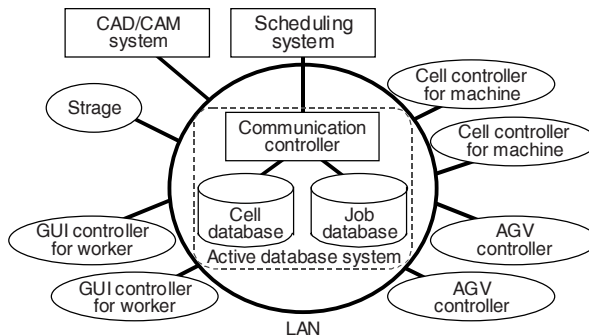


Fig. 1. System architecture with active database

change in the production plan which comes from outside of the active database, next is worker requirements such as working hours and taste, another one is alternation in cells or AGVs such as failures and recoveries, the last one is the discrepancy between the estimated operating time for the scheduling system and the actual one for the workers, cells and AGVs.

3. Dynamic Scheduling

A production plan is given to the active database as a trigger. The active database stores the production plan, which consists of the representation and number for each product in the job database, and delivers it to the chosen workers or cells who are appropriate for the required operation according to the cell database. Estimated operating times for the required operations are sent back to the active database from the workers via the GUI controllers and the autonomous cells via the cell controllers. A bunch of the estimated operating times, corrected, is sent to the dynamic scheduling system to deliver the operations to the appropriate workers or cells.

The process flow of dynamic scheduling is shown in Fig.2. When a trigger occurs, the active database searches the job database and finds the waiting operations. Then, the workers and the cells that are appropriate are selected for each operation. The workers and the cells are assigned to the operations according to their abilities stored in the cell database. Each operation is delivered to the selected worker or cell, and each worker or cell estimates the operating time. The operating time estimated by

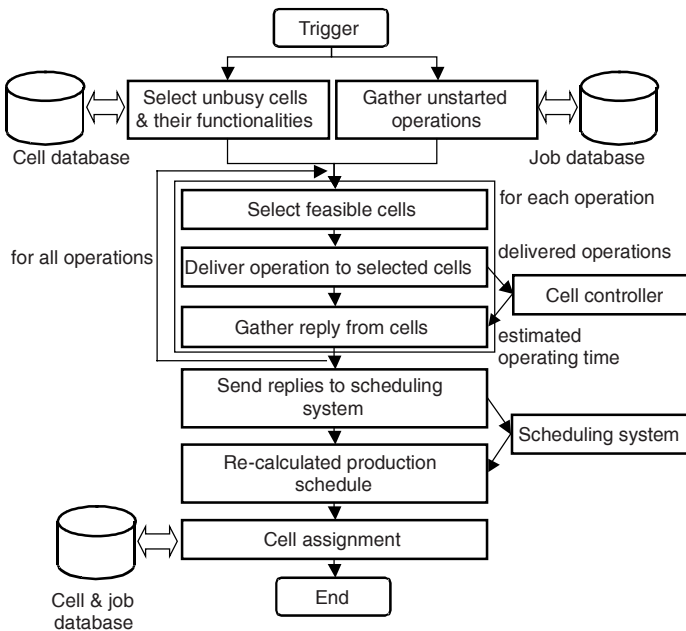


Fig. 2. Process flow of dynamic scheduling

each worker or cell is gathered by the active database. The gathered information is sent to the scheduling system to generate the newest production schedule. The scheduling system assigns the appropriate worker or cell for all operations under the condition that satisfies the shortest production time.

4. Considering Human Features

The dynamic scheduling mentioned above is one part of the architecture to realize a distributed production system. A distributed production system consists of autonomous cells or facilities, which cooperate with each other like human workers. That is why the dynamic scheduling can be applied to a human oriented production system. The biggest difference among autonomous facilities and human workers is that human workers have several features to be considered. For autonomous cells, it is enough to achieve the maximum productivity of the production system. But, for human workers, personal satisfaction must be considered beside productivity of the production system. The treatment of the features like satisfaction in the dynamic scheduling is a key issue in the human oriented production system. In this study, the following workers features are considered.

(a) Working hours.

Working hours of each worker can be decided by the worker him/herself. Working hours consist of normal hours and extra hours. Extra hours provide a better commission rate to workers. The worker can decide his/her working hours according to his/her life style and wishes.

(b) Financial award.

Financial award to each worker can be shown by the dynamic scheduling system. Usually, workers get salary partially by fixed income and partially by commission for certain results. In order to get more salary, they must extend their working hours and work harder. They can propose their own working hours independently according to their life style and expected salaries. The dynamic scheduling system assigns the jobs to the workers during their working hours, and shows the expected salary to the workers. Every worker can negotiate with the system interactively and select the jobs that satisfy them. But, the satisfaction of each worker is different, therefore it is impossible to generate an optimized production schedule which offers satisfaction to every worker.

(c) Friendship with co-workers and taste for jobs.

Friendship with co-workers and taste for jobs affects working ability. Good relationship with co-workers makes a good working environment, and the working ability is improved. Bad relationship with co-workers causes low working ability. On the other hand, taste for certain jobs affects the working ability, too. A good feeling about a favorite job results in good productivity and quality. The working ability affects the production schedule generated by the dynamic scheduling system. Friendship with co-workers and taste for jobs are defined for each worker in the cell database.

(d) Progressive skill.

The skill of each worker depends on his/her experience with the jobs, and affects working ability. The inexperienced workers have to be trained by the skilled workers through co-operation. Actually, the productivity of the production system decreases because the skilled worker is limited his/her ability because of co-operation with the inexperienced worker. But, their co-operation is needed to train the inexperienced worker and to maintain skill levels for the production system. As a result, the productivity of the production system improves because of the progressive skill of every worker. The default level of skill and the working hours needed to improve that skill are defined for each worker in the cell database.

(e) Flexibility of workers.

The flexibility of workers is a considerable feature. Human workers can handle several tasks and move freely in the factory. The system realizes the flexibility of human workers, and it achieves the dynamic reorganization of working groups. In this paper, the dynamic reorganization is verified in parallel YATAI production lines or parallel cell production lines.

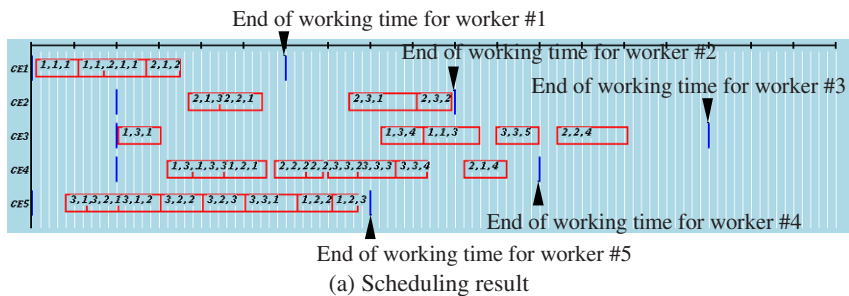
5. Case Studies

The distributed production system/human oriented production system treated in this paper starts to run according to the original production plan or schedule. The workers can send a trigger to the system whenever they want to change their working hours. When a trigger is sent, the active database starts to re-generate the production schedule referring to the working hours that are being updated and the working abilities that are being progressed. The negotiation time which is required for the re-generation of the production schedule is estimated. All jobs that would start within the negotiation time are assigned to the original workers to avoid interruptions in production.

Another worker who wants to change his/her working hours can also send a trigger to the system. Chains of triggers generate the modified production schedules, but these schedules are not guaranteed to be offer satisfaction to all workers.

Additionally, the system manager can send the trigger to change the original production schedule. It makes the dynamic reorganization of working groups possible. The suitable working groups to achieve the effective production will be generated using this function.

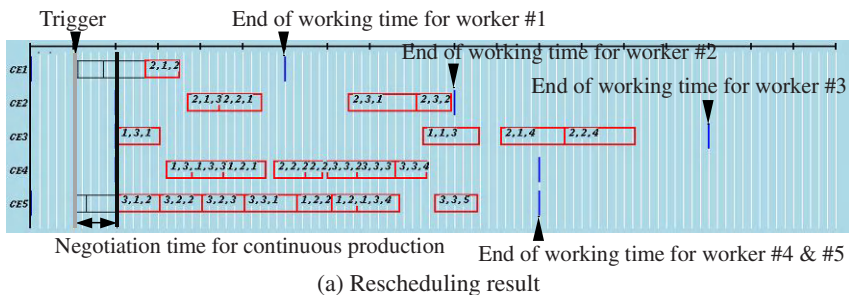
Figure 3 shows an original production schedule and the workers' awards. In Fig.3 (a), the small rectangles with numbers show the assigned jobs to the workers, and the short vertical lines show the start/end of their working hours. In Fig.3 (b), the workers' awards corresponding to their salary are shown as points estimated by the dynamic schedule system. When the worker corresponding to cell #5 wants to get more salary while he/she is working, he/she can send a trigger to inform the active database of the wish to extend his/her working hours.



CE 1	1st DAY		510 point
	Total		510 point
CE 2	1st DAY		431 point
	Total		431 point
CE 3	1st DAY		731 point
	Total		731 point
CE 4	1st DAY		722 point
	Total		722 point
CE 5	1st DAY		850 point
	Total		850 point

(b) Workers' award corresponding to their salary

Fig. 3. Original production schedule and workers' award



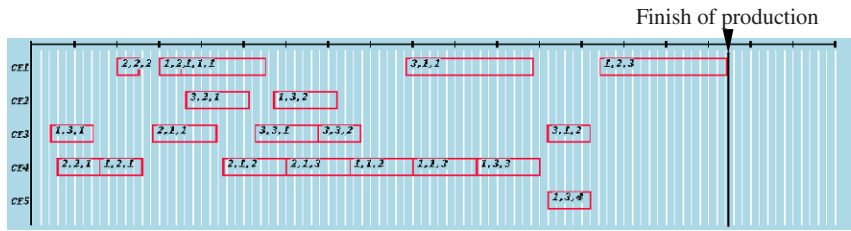
CE 1	1st DAY		510 point
	Total		510 point
CE 2	1st DAY		431 point
	Total		431 point
CE 3	1st DAY		711 point
	Total		711 point
CE 4	1st DAY		622 point
	Total		622 point
CE 5	1st DAY		1050 point
	Total		1050 point

(b) Workers' award corresponding to their salary

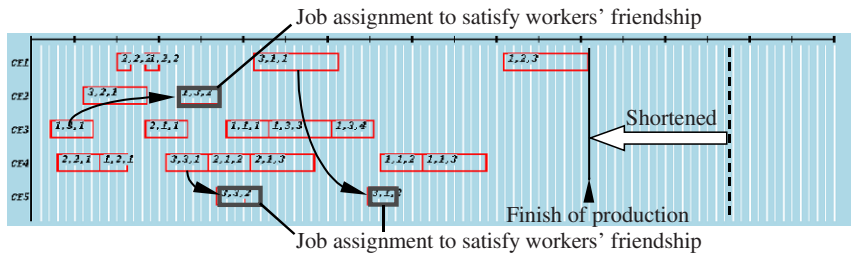
Fig. 4. Rescheduling results generated by request from worker #5

According to this trigger, the active database regenerates a new production schedule shown in Fig.4. The workers' award of cell #5 increases from 850 to 1150 as he/she wanted, but the workers' awards of other workers decrease as a result. Especially, the worker's award of cell #4 decreases from 722 to 622. After this rescheduling, the worker corresponding to the cell #4 can send a trigger to inform the active database of the wish to extend his/her working hours to keep his/her salary. Of course, other workers can send triggers to maintain their salaries and working satisfaction, as well.

Figure 5 shows the scheduling results in which the workers' friendship is considered and not considered. In the scheduling result shown in Fig.5 (a), the workers' friendship is not considered. In the scheduling result shown in Fig.5 (b), the workers' friendship is considered. In this case, the successive processes are assigned to the workers who have good relations with each other. The workers' friendship has a higher priority than the productivity to generate the production schedule. However, it is assumed that the workers' friendship generates a good feeling about work, and the good feeling improves the working ability, 1.5 times higher. As a result, the productivity is improved and the total production time becomes remarkably shorter. Also, the quality of the products will be improved owing to the good feeling about work. It is difficult to estimate the improvement of working ability, but good results will be obtained when workers' friendship and satisfaction is considered.



(a) Scheduling result unconsidered workers' friendship



(b) Scheduling result considered workers' friendship and improving ability

Fig. 5. Improvement of production schedule caused by workers' satisfaction

Figure 6 shows the scheduling results in which the co-operative jobs performed by the inexperienced worker and the skilled worker are both considered and not. In the

result shown in Fig.6 (a), the co-operative jobs are not considered, and only one job is assigned to the inexperienced worker corresponding to cell #5. In the result shown in Fig.6 (b), the co-operative jobs are considered, and the skilled worker corresponding to cell #1 trains the inexperienced worker through their co-operation. As a result, the total production time becomes longer, because the skilled worker is limited his/her ability by the co-operation with the inexperienced worker. However, their co-operation is needed to maintain skill levels for the production system. Otherwise, the inexperienced worker has no chance to improve his/her skill.

In the result shown in Fig.6 (c), the progressive skill of the inexperienced worker is considered. In this case, it is assumed that the progressive skill makes the working ability 2 times higher after several co-operative jobs. As a result, the productivity is improved and the total production time becomes remarkably shorter. It is difficult to estimate the improvement of the working ability, but good results will be obtained when the workers' progressive skill is considered.

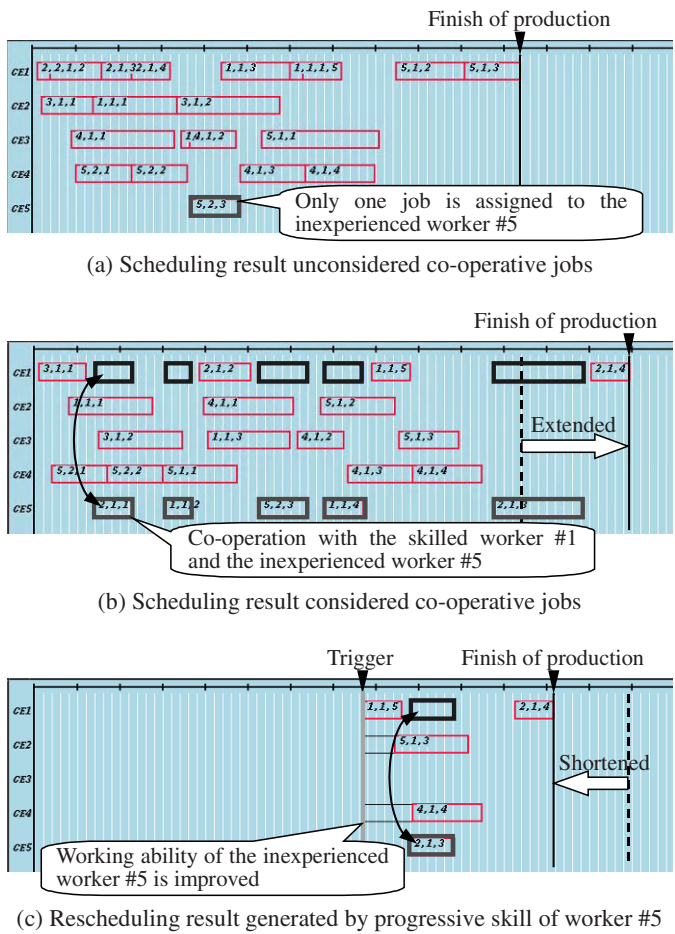
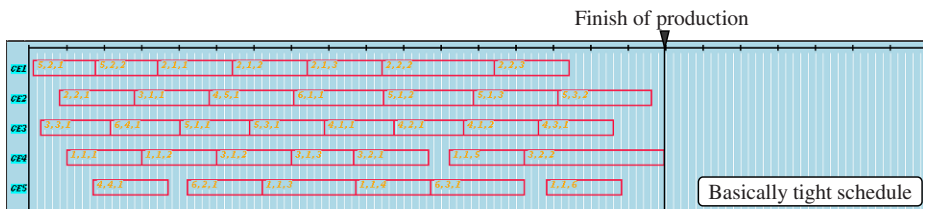
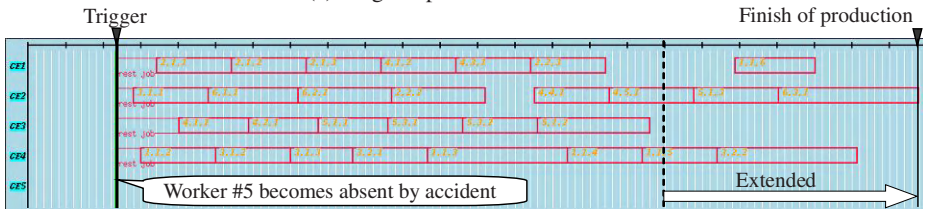


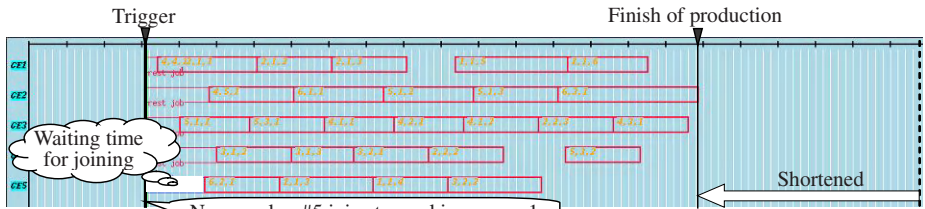
Fig. 6. Production schedule considered co-operative jobs and progressive skill



(a) Original production schedule

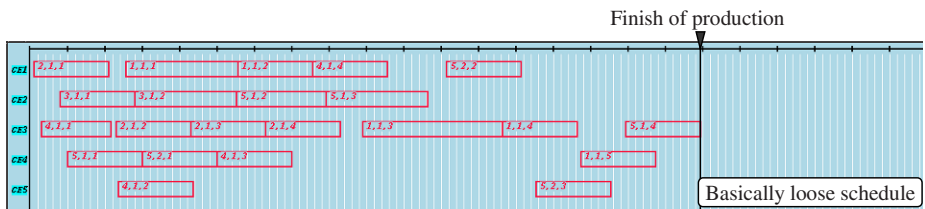


(b) Rescheduling result generated by accident of worker #5

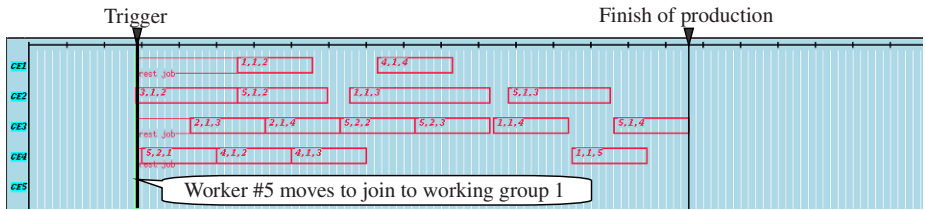


(c) Rescheduling result after dynamic reorganization, new worker #5 joins to working group 1

Fig. 7. Production schedules of working group 1 considered dynamic reorganization



(a) Original production schedule



(b) Rescheduling result after dynamic reorganization, worker #5 moves to join to working group 1

Fig. 8. Production schedules of working group 2 considered dynamic reorganization

Finally, the dynamic reorganization of working groups and the flexibility of workers is considered. Figure 7 and 8 show the scheduling results of before and after the dynamic reorganization of working group 1 and 2, respectively. The dynamic reorganization of working groups in parallel YATAI production lines or parallel cell production lines is treated in this paper. Basically, one production line consisting of working group 1 has a tight schedule as shown in Fig.7 (a). In this environment, the worker corresponding to cell #5 in working group 1 becomes absent by his/her accident, and a trigger occurs to re-generate the production schedule.

However, the total production time is extended remarkably as shown in Fig.7 (b). On the other hand, another production line consisting of working group 2 has a loose schedule as shown in Fig.8 (a). Therefore, the system manager sends another trigger to re-generate the production schedule, in order to make the worker corresponding to cell #5 in working group 2 join working group 1. The total production time after rescheduling is reasonable as shown in Fig.8 (b). After that, the system manager sends another trigger to re-generate the production schedule. This time, the new worker joins working group 1 from working group 2, and the total production time is shortened remarkably as shown in Fig.7 (c). In this scheduling, the waiting time for moving and joining group 1 from group 2 is considered.

It becomes possible to dynamically reorganize working groups by considering the flexibility of human workers. Appropriate working groups that achieve effective production will be generated using this function.

6. Conclusion

In this paper, a management architecture for a human oriented production system which fulfills both the effective utilization of production facilities and the satisfaction of workers or craftsmen is proposed. The proposed system runs as a centralized and distributed production system, and it generates a new production schedule interactively while considering the workers' satisfaction. In this system, machine trouble, machine repair or changes in the production plan are notified as triggers to generate the newest production schedule. Similarly, a change of working hours or a workers' disagreement are notified as triggers, and as a result the workers can get a new production schedule in order to satisfy them. Although the regenerated production schedules are not guaranteed to offer satisfaction to all workers, the proposed management architecture gives the opportunity to change the present schedule when the workers feel dissatisfied.

Technological subjects still remain; however, the human factors should be considered in the future flexible production system.

References

1. Arai, E. and Amunay, S. T.: An Approach to Flexible Cell Assignment and Dynamic Scheduling in FMS, *Systems Control and Information*, Vol.39, No.10, 1995, pp549-556.
2. Okino, N.: A Prototyping of Bionic Manufacturing System, *Proceedings of Int. Conference on Object-Oriented Manufacturing Systems*, 1992, pp297-302.

3. Ueda, K.: An Approach to Bionic Manufacturing Systems Based on DNA-Type Information, Proceedings of Int. Conference on Object-Oriented Manufacturing Systems, 1992, pp305-308.
4. Arai, E. et. al.: Auction Based Production System through Active Database System, Proceedings of the Pacific Conference on Manufacturing (PCM'96), 1996, pp371-375.

GeoCosm: A Semantics-Based Approach for Information Integration of Geospatial Data

Sudha Ram, Vijay Khatri, Limin Zhang, and Daniel Dajun Zeng

Department of MIS, University of Arizona, Tucson AZ 85721
{ram, vkhatri, lzhang, zeng}@bpa.arizona.edu

Abstract. Information integration implies access to distributed information sources without interfering with the autonomy of the underlying data sources. Integration of distributed geospatial data requires a mechanism for selecting the data sources and performing data processing operations on the selected sources efficiently. We describe a semantics-based information integration approach that uses a spatio-temporal semantic model to define the geospatial information content of the sources, employs a conflict resolution ontology to resolve semantic heterogeneity, and uses geospatial metadata to help the users evaluate usefulness of the available data sources. We show how the captured metadata can be used for efficient query planning. Based on our proposed approach, we are developing GeoCosm, a web-based prototype that would help integrate autonomous distributed heterogeneous geospatial data.

1 Introduction

Geospatial data is related to phenomena occurring above, on or below the Earth's surface, and is defined as having a *theme* (i.e., the phenomenon or object being observed), the *location* of the phenomenon and the *time* related to the phenomenon. The recent revolution in network connectivity and online availability of geospatial data has helped the Earth scientists, e.g., environmental planners and research hydrologists, access information from multiple data sources via the World Wide Web (WWW). However, the geospatial data required by these users are scattered among numerous agencies like USGS (United States Geological Survey), USDA (United States Department of Agriculture), EPA (Environmental Protection Agency), NOAA (National Oceanic and Atmospheric Administration), NASA EOSDIS (Earth Observing System Data and Information System) and various DAACs (Distributed Active Archive Center); as a result, the possible usage of available distributed geospatial data remains limited. Information integration implies access to multiple information sources such that the users are transparent to the distributed nature of the underlying data sources. In this paper, we describe a semantics-based approach to information integration that uses a spatio-temporal semantic model to define the information content of the geospatial sources, employs conflict resolution ontology to resolve semantic heterogeneity, and geospatial metadata that helps users evaluate usefulness of the available data sources.

Integration of heterogeneous distributed data requires [1]: (i) a mechanism for selecting the sources where different pieces of data may reside and (ii) performing

data processing operations in an efficient manner. The first problem (i.e., “where” is the data) is characteristic of distributed heterogeneous systems and the second (i.e., “how” to obtain the data) has been the center of research for traditional query optimization in databases. Mediators [22] provide a mechanism for resolving heterogeneity among distributed data. A global model of the application domain is used to define the information content of the sources. The user queries the resulting global schema without being concerned about the underlying data sources. On the other hand, query optimization requires a mechanism for selecting sources and ordering the operations to optimize query execution cost. In this paper, we propose information integration approach that employs *semantic*, *context* and *geospatial metadata* to efficiently process spatio-temporal queries. While semantic metadata specifies how the geospatial data is represented and organized in distributed sources, context metadata defines the logical relationships between the data in distributed data sources; on the other hand, geospatial metadata supports the end-user in assessing and using geospatial data.

Our work makes several contributions. (i) We have extended traditional information integration methodologies, e.g., [6, 8, 12], for geospatial data. The basis of any information integration approach is a canonical model used for describing the underlying information space. We employ a spatio-temporal semantic model [9, 10] that captures the semantics related to space and time. By providing a mechanism for representing the meaning of geospatial data, the spatio-temporal semantic model bridges the semantic gap between the data producers and its consumers. (ii) Closely associated with geospatial data is geospatial metadata that describes the geospatial data, e.g., processing performed on the geospatial data, sensor used for capturing geospatial data and data accuracy. Federal Geographic Data Committee (FGDC) [4] has proposed a common set of definitions of geospatial metadata that plays a role in *availability* (i.e., assess whether requested data is available), *fitness of use* (i.e., evaluate if the users’ specific need is met), *access* (i.e., determine the means of accessing the identified data) and *transfer* (i.e., procedure for processing and using the downloaded data) of data; agencies collecting geographic data are mandated to maintain geospatial metadata using FGDC standard. Our architecture employs geospatial metadata—as specified by FGDC—that helps users evaluate the usefulness of geospatial data, and understand how to access and process the available data. (iii) We show how the captured metadata can be employed for efficient query planning, which is essentially a partially ordered set of relational operators. Our approach extends the Planning by Rewriting (PbR) framework [1] by interleaving plan generation and plan execution [20], thus, addressing the issues of very large datasets and dynamic information sources associated with geospatial data.

The rest of the paper is organized as follows. In section 2, we describe a motivating example. In section 3, we summarize two key components of our architecture: metadata management and query plan generation. We describe our proposed architecture in section 4. Based on the architecture described in section 4, we describe a proof-of-concept prototype system we are developing and how it can be used to solve some of the problems described in section 2. Finally, comparison with related work and future research directions round out this paper.

2 Motivation

To exemplify the problems associated with using distributed geospatial data, let us take a motivating example of runoff modeling associated with floods. In the United States, about 7 percent of the land area is subject to flooding with losses amounting to over \$2.5 billion in direct damages [3]. The researchers need to evaluate the runoff response to rainfall to evaluate flood risks and ensure hazard understanding, preparedness and response. However, the quality of run-off models and the predictions based on these models are dependent on data that forms an input to these models. Some of the inputs to runoff modeling include snow depth, snow covered area (SCA), snow water equivalent (SWE), precipitation, vegetation type and relative humidity.

There are many challenges associated with using distributed geospatial data for runoff modeling described above. (i) *Distributed and Heterogeneous Data*: Typically geospatial data required by the end-users is collected by a large number of agencies and users do not even know where the required data may be available. For example, federal agencies collecting precipitation data itself include U. S. National Weather Service (NWS), U. S. Army Corp of Engineers, USDA Soil Conservation Service, the USDA Forest Service, the Bureau of Reclamation and Tennessee Valley Authority. (ii) *Autonomous Data Sources*: Distributed geospatial data sources are independently managed, support a wide range of querying capabilities, and have different types of user interfaces. This necessitates that users be cognizant of multitude interfaces and capabilities of different sources. (iii) *Identifying Data Relevant for the User*: Even if the users do find the data source, they spend inordinate time trying to identify if the data is relevant for their needs [16]. Closely associated with geospatial data is geospatial metadata (e.g., spatial resolution, projection and temporal extent) that helps users evaluate usefulness of the geospatial data. (iv) *Very Large Datasets*: Typically geospatial data required by users may be of the order of megabytes. This implies that downloading data to discover that it does not serve the purpose of the user would be wasteful both from the user's (i.e., waste of time) and data provider's (i.e., waste of processing resources) perspective. (v) *Dynamic Data Sources*: Typically spatio-temporal data sources are temporal with large amount of data added everyday. For example, data arriving from the *Mission to Planet Earth Satellites* will grow by 5 terabytes a day and is expected to grow to 15 petabytes by 2007 [7].

Based on the challenges described above, there are many users' requirements. (i) *Access to distributed sources through a single integrated system*: The users should be able to make queries while being transparent to the distributed nature of underlying data sources. We employ a spatio-temporal schema that defines the information content and acts as a "window" to the underlying distributed geospatial datasets; this is referred to as a *global schema*. The content of the distributed information sources (referred to as *local schemas*) are then mapped with the global schema. Our approach that only stores the semantic metadata and the mapping information in a central repository ensures that our architecture is scalable and can be used to integrate a large number of data sources. (ii) *A mechanism to automatically resolve semantic heterogeneities*: To ensure exchange of heterogeneous data, there is a need for a technique to resolve semantic conflicts among the information sources. Ontology is a representational vocabulary for a shared domain of discourse and is necessary for

information integration as it are used to describe the data semantics of information sources. (iii) *Mechanism for identifying the relevance of available data*: Geospatial metadata is indispensable in the process of locating data and assessing its fitness of use for the end-user. We employ geospatial metadata proposed by FGDC [4] that would allow users to assess and evaluate if the geospatial data fits their requirements. Thus, geospatial metadata helps prevent waste of time in downloading irrelevant datasets, and helps reduce the associated transaction and network costs. (iv) *Query Response Time*: This requirement necessitates the ability to monitor and react to changes across networked sources and the capability to adapt to changing environment with respect to data availability. We employ metadata and dynamic status of the environment for efficient query planning.

3 Information Integration Approach

In this section, we describe two key components of our architecture: (i) metadata management that provides a mechanism to define “where” is the data and (ii) query plan generation that describes “how” to obtain that data.

3.1 Metadata Management

Our approach to metadata management includes a spatio-temporal semantic model, ontology and geospatial metadata.

3.1.1 Spatio-Temporal Semantic Model

A semantic model provides a way to conceptually define the *miniworld* and captures the information content of the underlying data sources. We employ ST USM (Spatio-Temporal Unifying Semantic Model) [9, 10] as a canonical model to define the spatio-temporal information space. ST USM, based on USM [15], provides a mechanism to capture the semantics related to space and time, e.g., *event* and *state*, *valid time* and *transaction time*, *existence time*, *temporal granularities*, *valid time indeterminacy*, *geometry* and *position*, *spatial resolution* and *imprecision*, and *change in position* and/or *shape* over time. We summarize the temporal and spatial aspects that need to be represented for geospatial applications.

Time-varying data may be modeled as an event or a state. While an event occurs at a point of time (i.e., an event has no duration), a state has duration, e.g., rainfall from 5:12 PM to 5:46 PM was 1.2 inches. Facts can interact with time in two orthogonal ways resulting in transaction time and valid time. Valid time denotes when the fact is true in the real world and implies the storage of histories related to facts. On the other hand, transaction time links an object to the time it is current in the database and implies the storage of versions of database objects. Existence time, which applies to objects, is the valid time when the object exists. Temporal granularity (or resolution) is the measure of temporal (or spatial) datum, e.g., *gregorian-day* and *business-week* (or *degree*). Indeterminacy (or imprecision), or “don’t know *exactly* when or where”, is related to granularity and may be pertinent for some database applications.

A spatial object is associated with geometry and position. Geometry represents the shape and size of an object. The position in space is based on co-ordinates in a mathematically defined reference system, e.g., latitude and longitude. Geometry of the spatial object may be 0-, 1- or 2- dimensional corresponding to a *point*, a *line* and a *region*. A point is “a zero-dimensional spatial object with co-ordinates”, a line is “a sequence of ordered points, where the beginning of the line may have a special start node and the end a special end node” and a region or polygon consists of “one outer and zero or more inner rings” [21].

For example, avhrr (source agency, measure-date, geo, depth, SWE) represents a global schema where SWE and snow depth are measured by a source agency. SWE and depth are time varying spatial attributes. In other words, *spatial projection* on these attributes is defined as $\pi^y(\text{SWE}) = \text{geo}$, where geo represents *points* on the surface of the Earth “where” SWE is measured, i.e., *position*. Similarly, *valid time temporal projection* is defined as $\pi^t(\text{SWE}) = \text{measure-date}$, where measure-date represents an *event* “when” SWE was measured, i.e., *valid time*. Additionally, let us assume that our runoff modelers are interested in the Colorado Basin, i.e., *within* (Colorado Basin, *geo*), where *within* (g_1, g_2) is a spatial predicate implying that g_2 is ‘spatially within’ g_1 [14]. Let us assume that this global schema integrates two remote data sources, referred to as source1 and source2. Source1 includes snow-depth and Snow-Water-Equivalent in the Gunnison Basin, i.e., source (date, snow-depth, geo, Snow-Water-Equivalent):- avhrr (source agency, measure-date, geo, depth, SWE) and *within* (Gunnison Basin, *geo*). On the other hand, source2 includes depth-of-snow and SWE for Lower Green Basin from 1997 to 1998, i.e., source (timestamp, depth-of-snow, SWE, geo):- avhrr (source agency, measure-date, geo, depth, SWE) and *within* (Lower Green Basin, *geo*) and $1997 \leq \text{measure-date} \leq 1998$. These schemas, referred to as local schemas, represent the information content of the remote sources.

3.1.2 Ontology

Ontology is defined as a representational vocabulary for a shared domain of discourse. Formally represented knowledge is based on *conceptualizations* [13], which consist of objects and relationships that hold between them. We represent ontology as a tree where $\forall \text{ child-concept}, \exists \text{ parent (child-concept)} = \text{parent-concept}$. For example, distance is an example of parent-concept and inch and centimeter are examples of its child-concept; on the other hand, distance is a child-concept of data unit. Additionally, one can define functions between leaf nodes, i.e., *sibling-relation* ($\text{concept}_1 = \text{concept}_2 \Rightarrow (\forall i, \nexists \text{ concept}_i, \text{parent}(\text{concept}_i) = \text{concept}_1 \vee \text{concept}_2) \wedge \text{parent}(\text{concept}_1) = \text{parent}(\text{concept}_2)$); this implies that functions can only be defined between leaf concepts with same parents, i.e., siblings. For example, inch-to-cm (inch) = 2.54-centimeter; in this example, inch and centimeter are leaf concepts of parent distance. An ontology repository, referred to as context metadata, stores concepts and functions between leaf concepts in the ontology tree (e.g., between centimeter and inch).

Ontologies are necessary for information integration from multiple heterogeneous information sources. For example, source₁ may have snow-depth in centimeter and while source₂ may have depth-of-snow in inch. The local schema-ontology association, referred to as *context mapping*, helps resolve semantic heterogeneity in

the distributed data sources. Formally, $\text{distance}_r(\text{snow-depth}) = \text{centimeter}$ and $\text{distance}_r(\text{depth-of-snow}) = \text{inch}$. This explicit representation of context associated with remote data can be employed to resolve semantic heterogeneity and translate the context of a user query on the fly (e.g., provide snow-depth to user in inch, centimeter or meter, as desired by the user).

3.1.3 Geospatial Metadata

Geospatial metadata is indispensable in the process of locating information and evaluating its fitness for specific purposes. The intent of geospatial metadata is to allow anyone not intimately involved in data collection, compilation or production efforts to make sense and effective use of the data. FGDC has developed Content Standard for Digital Geospatial Metadata [4] related to content, quality and other characteristics of geospatial data. This standard formally defines elements—over 200—known as standard elements. From these elements, we have incorporated several elements from seven sections: *Identification Information* (i.e., basic information about dataset, e.g., abstract and purpose), *Data Quality* (i.e., assessment of data quality, e.g., positional accuracy and processing step), *Spatial Reference* (i.e., a means to encode coordinates of datasets, e.g., geographic coordinate units), *Metadata Reference* (i.e., information on currentness of metadata), *Citation* (i.e., recommended reference), *Time Period* (i.e., date and time of event) and *Contact Information* (i.e., the identity of person/organization associated with the dataset). The geospatial metadata associated with remote sources helps the users assess the usefulness of data. For example, the access constraints and use constraints (part of Identification Information) of source would have implications on the potential use of data at source.

3.2 Query Plan Generation

A query plan is a partially ordered set of relational operators. The efficient execution of a plan depends on information sources that are selected, data operations that need to be executed on the selected sources, and the execution order of these operations. In this section, we first overview spatio-temporal queries and then describe our approach to query plan generation; we describe details related to query plan elsewhere [17].

Continuing with our motivating example, a user may give a spatio-temporal query like “What was snow water equivalent within Colorado Basin between May 15, 1998 and May 30 2000”. This query is an example of *theme projection* [18] with signature $\text{theme} \times \{A_1, \dots, A_n\} \rightarrow \text{theme}$. If *geo* is the spatial type and *T* is an instance of a theme and *A*₁, *A*_{*n*} are its descriptive attributes, theme projection may be denoted by $\pi_{A_1, \dots, A_n, \text{geo}}(T)$. Similarly, the signature of theme selection is $\text{theme} \times p_{A_i} \rightarrow \text{theme}$, where p_{A_i} is a predicate on descriptive attributes. Thus, our query may be written as $\pi_{(\sigma_{\text{within}(\text{r}, \text{Si}, \text{geo})} \wedge \text{SUR t between and 2000})}(\text{avhrr})$ where *avhrr* represents the snow theme. Based on topological relationships, various spatial predicates between regions include, e.g., *equals* (g_1, g_2), *disjoint* (g_1, g_2), *intersects* (g_1, g_2), *touches* (g_1, g_2), *within* (g_1, g_2), *contains* (g_1, g_2), *overlaps* (g_1, g_2), where g_1 and g_2 represent geometries of the regions.

Our approach of query plan generation is an extension of prior work, particularly, the PbR [1] framework, which is designed to address plan efficiency and plan quality. The framework first generates an initial solution plan, then iteratively rewrites the current solution plan in order to improve its quality (in terms of execution cost) using a set of declarative plan rewriting rules until either an acceptable solution is found or a resource limit is reached. Our framework includes: (i) pre-planning for dynamic factors, (ii) rule-based plan rewriting, and (iii) interleaving of plan reformulation, and plan execution. To ensure system response to the changing environment, the pre-planning process uses real-time data such as current network speed and source status to model the dynamic environment. Additionally, geospatial metadata (i.e., minimum bounding rectangle formed by *westbc*, *eastbc*, *northbc* and *southbc* [4]) is employed to choose sources that “roughly” satisfy the spatial query; this helps eliminate the sources that will not be useful for subsequent evaluation of (very expensive) spatial predicates. Our rule-based plan rewriting is similar to the PbR [1] approach, where the system iteratively applies optimization and restriction rules to an initial sub-optimal plan. Rules are used to rewrite the current partial plan, e.g., *selection first*, *selection swap*, and *join union distribution*. Interleaving of planning and plan execution can be described as a recursive process that decomposes the plans into subplans, executes subplans to retrieve partial results, generates alternative plans and replans. We have adopted the framework proposed in [20] to decompose and execute the query plan.

4 Architecture

As shown in Fig. 1, our architecture involves three key components: metadata manager, mapping manager, and query manager.

As a precursor to the user querying the remote data sources via a single conceptual interface, the associated metadata needs to be captured. *Metadata Manager* provides a mechanism for explicit representation of spatio-temporal semantics (*semantic metadata*), the use of ontology for capturing the data context (*context metadata*), and *geospatial metadata* for assessing the usefulness of the data sources. We employ ST USM as the canonical model for information integration.

Next, the local schema is mapped with the global schema (via *semantic mapper*) and with ontology concepts as specified in context metadata (via *context mapper*). The objective of the *mapping manager* is consolidation of the distributed data sources with a common model for query processing. The context mapper captures the fundamental assumptions about data explicitly, and describes the meaning the real world concepts represented in the information systems.

The user interface interacts with the *query manager*, where the *mediator* uses metadata and mapping information to dynamically interleave a *query plan* with query execution. The first step of the mediator is to select appropriate sources for the query and generate an initial query plan based on mapped metadata and dynamic status of the distributed data sources. Next, the mediator iteratively applies the rewriting rules to the query plan to improve its quality, and decompose the plan into sub-plans. The last step is to execute the executable sub-plans, and go back to step 1 or 2 until all the sub-plans have been successfully executed and final results have been generated.

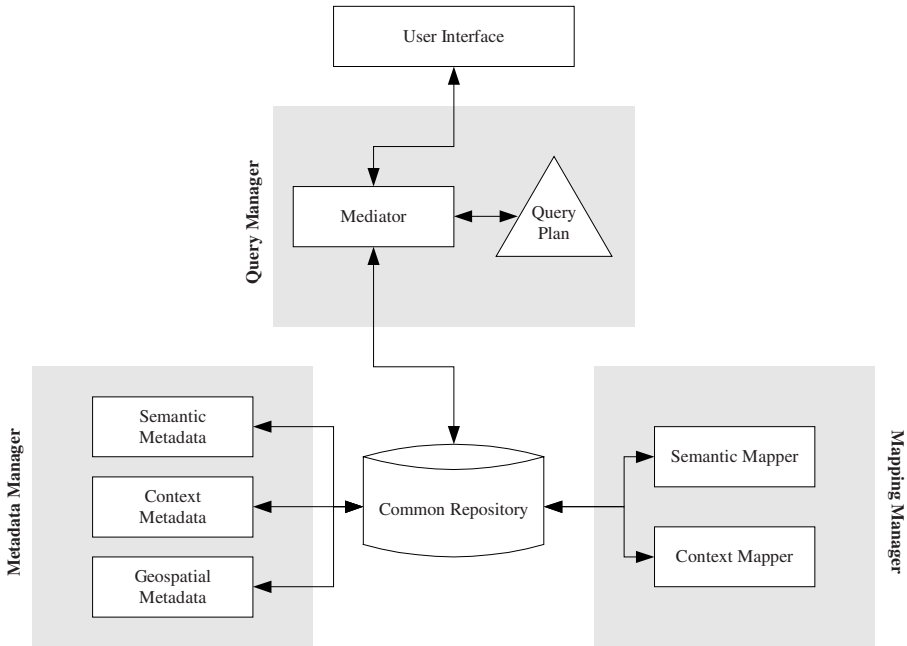


Fig. 1. GeoCosm Architecture

Interleaving of query plan and plan execution helps generate scalable high-quality plans that take dynamic environment into consideration.

5 GeoCosm: A Web-Based Prototype

Based on our approach described in the previous section, we are developing a proof-of-concept prototype called GeoCosm. GeoCosm provides an interface for: (i) an expert to register a source by entering the associated metadata; and (ii) users to query the distributed data sources through a single conceptual interface.

First the expert registers a source via semantic mapper (Fig. 2), enters geospatial metadata (Fig. 3) and the associated context mapping (not shown). In Fig. 2, the expert enters the source name (`source1`) and associates the remote schema with the global schema. The annotation phrase “E(second)/-/P(dms-deg)/P(dms-deg)/-/” associated with SWE succinctly represents a time-varying spatial attribute. The temporal and spatial annotation are segregated by “/”. The temporal annotation first specifies that SWE is an event (E) captured at the temporal granularity of second (sec) and that the application does not need to capture transaction time (i.e., “-” after “/”). The spatial annotation is specified for the x-, y- and z-dimensions each segregated by “/”, and indicates that SWE is represented as point (P) on the surface of the Earth with spatial granularity of dms-deg (degree-minute-second); as SWE is specified only for points on the surface of the Earth (i.e., the xy-plane), the z-dimension geometry is specified as

“-”. Details related to the semantics of spatio-temporal annotations are outside the scope of this paper and described in detail elsewhere [9, 10].

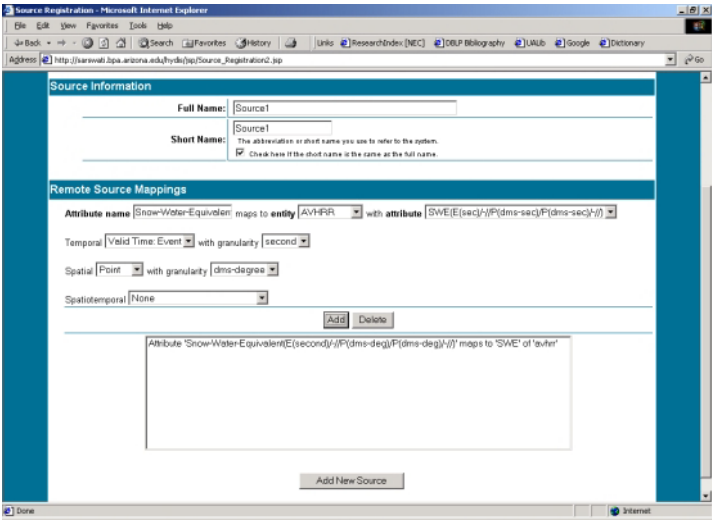


Fig. 2. Semantic Mapping

Next the domain expert enters the geospatial metadata. As described in a previous section, we employ geospatial metadata as specified by FGDC [4]. For example, in Fig. 3 (geospatial metadata in XML) westbc, eastbc, northbc and southbc are defined by FGDC to specify the minimum-bounding rectangle for the geospatial datasets.

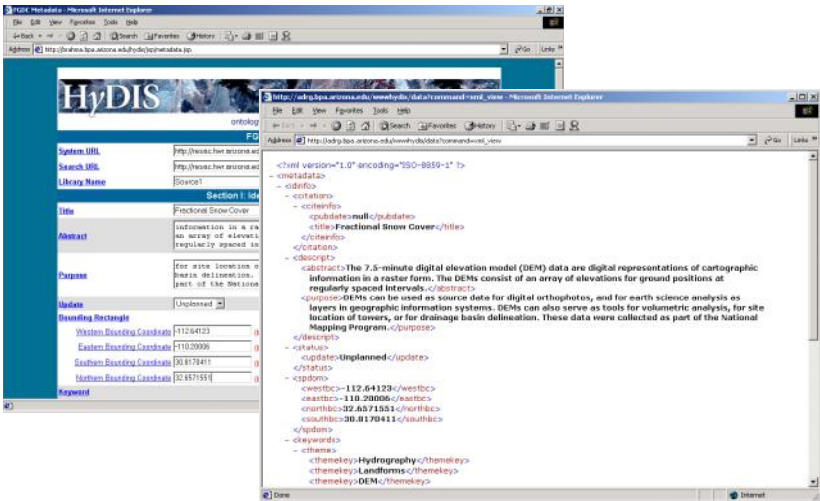


Fig. 3. Geospatial metadata

Having described source registration, we next describe the query interface. The end user interacts with an interface where they can give a spatio-temporal query. For example, in Fig. 4 the end user can select source, categories, date (starting date and ending date), and specify the region (basin); in other words they can specify the theme, time and region. The data displayed on this screen is based on the metadata entered by the expert previously. For example, the two sources appear because metadata related to source1 and source2 has been registered previously. Continuing with our example, our user is interested in flood modeling and needs SWE related to snow within Colorado Basin between May 15, 1998 and May 30, 2000.

Fig. 4. Geospatial Query Interface

Based on the query specified (e.g., Fig. 4), the user is shown preliminary results (Fig. 5). At this point, the user may get many “hits” and the user may want to perform comparison of the preliminary results. Typically, each of these datasets is very large in size and users spend inordinate time trying to figure if the data serves their purpose. The user can specify the criteria of comparison (based in FGDC metadata) and “checks” the pertinent criteria (e.g., identification in Fig. 5). On clicking the compare button, the user is shown, e.g., Fig. 6, based on which she can make a better decision as to whether the data would be useful for their requirements. For example, the user is shown two datasets with different associated dates (1/12/1998 and 05/20/2000), and different regions (Gunnison Basin and Lower Green Basin) within Colorado Basin. Additionally, Identification Metadata (e.g., update and access constraints) can help the user choose the data that best serves their needs. The user can save the link to the dataset using “Save” button and may continue with further search on other themes related to run-off modeling.

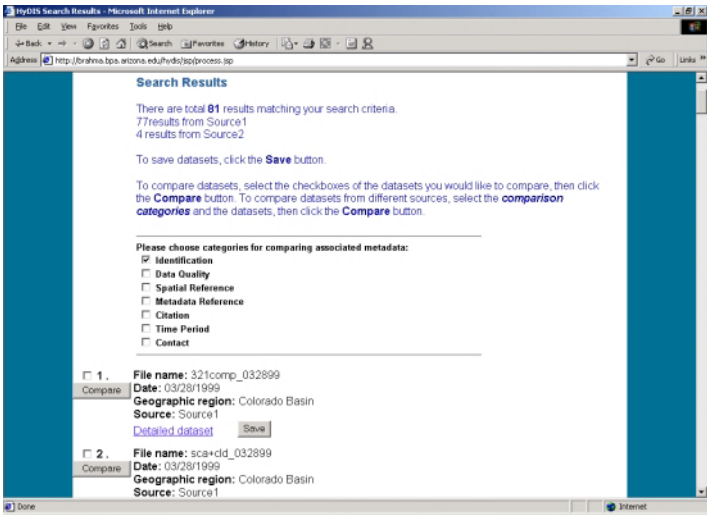


Fig. 5. Preliminary Query Results

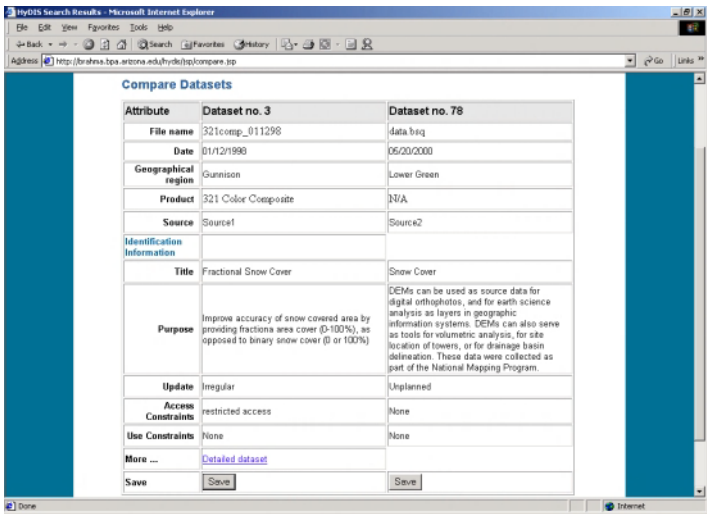


Fig. 6. Comparison of Query Results

In this section, we described GeoCosm—an information integration prototype we are developing—which provides a mechanism to capture metadata (e.g., Fig. 2 and Fig. 3) and presents an integrated conceptual interface where the user can query distributed spatio-temporal datasets, compare the results based on geospatial metadata and choose the ones that best serves their needs.

6 Related Work

Various techniques for integration of distributed data have been presented, which can be classified as *declarative* (e.g., [2, 19]) and *procedural* (e.g., [5, 11]). In declarative approach, a wrapper translates the domain schema from local data model to a system wide canonical model. An integrated global schema is developed and there may be a single global schema or multiple global schemas for different class of users. Mediator combines multiple local schemas and resolves inconsistencies. On the other hand in the procedural approach, the end-user is not provided with a pre-defined view and they access the metadata dictionary that contains the description of local schemas. Using the metadata, the end-user creates an integrated schema on the fly and poses queries against the custom-built view. In this approach, it is assumed that the schemas can evolve and that end-user understands the semantics of the local schemas. The extant information integration approaches do not take into consideration the peculiarities associated with spatio-temporal data, e.g., spatial predicates are very expensive operations. Additionally, existing query-planning techniques assume static environment, i.e., all the inputs required for query plan are known a priori.

Our information integration approach may be considered as declarative and is based on the assumption that the users are not cognizant of the semantics of the underlying distributed data sources. We have extended traditional information integration methodologies for geospatial data. We employ a spatio-temporal semantic model as canonical model to capture the spatio-temporal semantics of the application. Additionally, we employ geospatial metadata that helps users evaluate the usefulness of data. We have extended the extant query planning approaches with an open world assumption, i.e., for problems where all the inputs required for a query plan are not known a priori.

7 Conclusions

Information integration is a core problem in distributed databases, cooperative information systems and data warehouses. In this paper, we described an approach to integrating spatio-temporal distributed data that employs metadata to efficiently generate query plans. GeoCosm is a part of the Hydrology Information Systems (HyDIS), an overall information system to provide comprehensive decision support and access to distributed heterogeneous hydrologic information sources. GeoCosm is being implemented using Java 2 (JDK 1.2) and Oracle 8.1.6 and can be accessed through a web-browser. The prototype system is currently running on NT workstations.

We are completing various aspects of GeoCosm. We are implementing the context metadata (and context mapper) that will help in resolving semantic heterogeneities in the distributed data. We are also implementing algorithms that will integrate query planning with query execution; next, we plan to evaluate our query plan generator with other related approaches. We are also extending our methodology in various directions. In future, we plan to augment our mapping manager with mapping rules verifier that will employ knowledge discovery techniques to discover discrepancies.

We believe that our approach to information integration can be used to integrate large number of distributed geospatial data sources, thus, helping improve the predictive power of the models based on distributed geospatial data.

Acknowledgments. The first three authors were supported in part by HyDIS, a NASA-funded project.

References

- [1] J. L. Ambite and C. A. Knoblock, "Flexible and Scalable Query Planning in Distributed and Heterogeneous Environments," in Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems, Pittsburgh Pennsylvania, USA, pp. 3-10, 1998.
- [2] R. J. Bayardo, B. Bohrer, R. S. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. H. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, and D. Woelk, "The InfoSleuth Project," *SIGMOD Conference*, 543-545, 1997.
- [3] K. R. Bradbury, V. R. Baker, A. P. Barros, M. E. Campana, K. A. Gray, C. T. Haan, D. H. Moreau, C. L. Paulson, S. S. Schwartz, L. Shabman, K. D. Thompson, and D. A. Woolhiser, *Hydrologic Hazards Science at the US, Geological Survey Commission on Geosciences, Environment and Resources*: National Research Council, National Academy Press, 1999.
- [4] FGDC, "Content Standard for Digital Geospatial Metadata," , vol. 2001, FGDC-STD-001-1998 ed National Spatial Data Infrastructure, http://www.fgdc.gov/standards/documents/standards/metadata/v2_0698.pdf, 1998.
- [5] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom, "The TSIMMIS Approach to Mediation: Data Models and Languages," *Journal of Intelligent Information Systems*, Vol. 8, No. 2, pp. 117-132, 1997.
- [6] M. R. Genesereth, A. M. Keller, and O. M. Duschka, "Infomaster: An Information Integration System," in The ACM SIGMOD International Conference on Management of Data, Tucson, AZ, pp. 1997.
- [7] J. Gray, "Evolution of Data Management," *IEEE Computer*, Vol. 29, No. 10, pp. 38-46, 1996.
- [8] M. A. Hearst, A. Y. Levy, C. Knoblock, S. Minton, and W. Cohen, "Information Integration," *IEEE Intelligent Systems*, Vol. 13, No. 5, pp. 12-24, 1998.
- [9] V. Khatri, S. Ram, and R. T. Snodgrass, "ST-USM: Bridging the Semantic Gap with a Spatio-temporal Conceptual Model," *TIMECENTER Technical Report TR-64* 2001.
- [10] V. Khatri, S. Ram, and R. T. Snodgrass, "Supporting User-defined Granularities and Indeterminacy in a Spatiotemporal Conceptual Model," *Annals of Mathematics and Artificial Intelligence (Also available as TIMECENTER Technical Report TR-55)*, 36 pages, forthcoming.
- [11] H. Kitajima, R. Masuoka, and F. Maruyama, "Integrating Information and Knowledge with Software Agents," *Fujitsu Science Technology*, Vol. 36, No. 2, pp. 162-174, 2000.
- [12] C. A. Knoblock, S. Minton, J. L. Ambte, N. Ashish, I. Muslea, A. G. Philpot, and S. Tejada, "The Ariadne Approach to Web-based Information Integration," *International Journal of Cooperative Information Systems*, Vol. forthcoming, 1-26, 2001.
- [13] N. F. Noy and C. D. Hafner, "The State of the Art in Ontology Design: A Survey and Comparative Review," *AI Magazine*, Vol. 18, No. 3, pp. 53-74, 1997.
- [14] OGIS, "OpenGIS Simple Feature Specification for SQL," Open GIS Consortium, Inc. 99-049, May 5 1999.
- [15] S. Ram, "Intelligent Database Design using the Unifying Semantic Model," *Information and Management*, Vol. 29, No. 4, pp. 191-206, 1995.

- [16] S. Ram, V. Khatri, Y. Hwang, and S. R. Yool, "Semantic modeling and decision support in hydrology," *Photogrammetric Engineering and Remote Sensing*, Vol. 66, No. 10, pp. 1229-1239, 2000.
- [17] S. Ram, D. D. Zeng, V. Khatri, and L. Zhang, "Efficient and Scalable Query Planning in Distributed Open Environments," in 11th Workshop on Information Technologies and Systems, New Orleans, Louisiana, pp. 2001.
- [18] P. Rigaux, M. O. Scholl, and A. Voisard, *Spatial Databases: With Application to GIS*: Morgan Kaufmann Publishers, 2001.
- [19] V. S. Subrahmanian, S. Adali, A. Brink, R. Emery, J. J. Lu, A. Rajput, T. J. Rogers, R. Ross, and C. Ward, "HERMES: Heterogeneous Reasoning and Mediator System," , vol. 2000 University of Maryland, <http://www.cs.umd.edu/projects/hermes/publications/abstracts/hermes.html>, 1997.
- [20] K. Sycara and D. Zeng, "Coordination of Multiple Intelligent Software Agents," *International Journal of Cooperative Information Systems*, Vol. 5, No. 2-3, pp. 1-15, 1996.
- [21] J. W. van Roessel, "Design of a Spatial Data Structure using the Relational Normal Form," *International Journal of Geographical Information Systems*, Vol. 1, No. 1, pp. 33-50, 1987.
- [22] G. Wiederhold, "Mediators in the Architecture of Future Information Systems," *IEEE Computer*, Vol. 25, No. 3, pp. 38-49, 1992.

Imposing Modeling Rules on Industrial Applications through Meta-modeling

Peter Fröhlich, Zaijun Hu, and Manfred Schoelzke

ABB Forschungszentrum, Wallstadter Strasse 59
68526 Ladenburg, Germany
{peter.froehlich, zaijun.hu, manfred.schoelzke}@de.abb.com

Abstract. There is a trend in the automation domain to create common conceptual models of an industrial plant, which serve different applications, like online displays, simulators, control systems, diagnostic tools and others. To master the complexity of such common models, modeling rules have to be enforced. Current solutions to this problem either rely on an informal definition of semantics or the semantics of one dedicated tool or language. In the current paper, we introduce a meta-modeling approach to provide and enforce such modeling rules. Our approach concisely defines consistency of conceptual models with respect to a meta model. We show how the approach is mapped to widely accepted general-purpose IT standards (like UML and XML) and tools.

1 Introduction

There is a trend in the automation domain to create common conceptual models of an industrial plant, which serve different applications, like online displays, simulators, control systems, diagnostic tools and others. Each application has its requirements on the contents of the conceptual model. For example, a simulator has to access the mathematical models for the different types of processes, the instances of these processes, their connections, etc. A graphical display has to access the icons for the different facilities and processes, the variables to be displayed and their types, etc.

To ensure, that a model meets such requirements and to allow the different applications to explicitly access the different categories of information, we structure our model into a meta model, a domain model and the project data (the instances of the classes in the domain model). The meta model defines the structure of the domain model. As we will see, the classes of the meta model serve as categories for the domain classes and the associations within the meta model define, which associations may exist within the domain model. All three layers of the model are explicitly represented using XML documents and a common XML schema [23, 3].

The paper is organized as follows: In section 2 we review the related work on meta modeling as well as currently available standards for representing models in application-independent formats. Section 3 describes our meta-modeling approach. It first shows, using a small example, how meta-models are used to specify domain model structures. Then defines mathematically when a domain model is a consistent instance of a meta model. Next the meta modeling approach is compared to patterns [11, 8], which are used to achieve

similar goals but have a different semantics. In section 4 we present the mapping of our formalism to an XML schema and XML documents. Section 5 concludes the findings of the paper.

2 Related Work

2.1 Modeling Formalisms

Meta Modeling is an important technique, which has been used to define general-purpose modeling languages like the UML [6]. Meta-Modeling has also been used to describe application domains, e.g. Requirements Engineering [22], Hypermedia Systems [9], or Software Configuration Management [10]. Recently, various papers describe the use of meta modeling in industrial applications, e.g. in the process industry [1, 16, 2]. Witness, that in some of the industrial application papers the term meta-model is used in a different sense compared to its IT meaning. In [1] the term model is used for a (mathematical) model of one piece of equipment, the term meta-model is used for an application-independent representation of a piece of equipment.

Powerful languages and systems are available for conceptual modeling. Within these, formal logic-based approaches to meta-modeling have been studied, e.g. [17, 13]. These formalisms with their rigorous semantics have been used to model application domains, as mentioned above. The expressive power of the logical formalisms however inherently leads to a limited size of databases, which are tractable. Due to these efficiency (and sometimes decidability) problems and the difficult concepts underlying the languages, they are not widely used in industrial applications.

Other approaches like [2, 6] use meta models as reference mainly for human readers. A formal definition of the semantics of the meta models is not given. Many initiatives are striving to capture the semantics of the UML meta model more formally [4, 7, 14]. The meta modeling formalism presented in the current paper has a limited scope. It serves the definition of modeling rules, which constrain the possible structures allowed in the domain model. Patterns have been used to achieve a similar goal. They have become popular after [11] and have also been used for conceptual modeling [8]. We discuss the differences between our meta-modeling approach and a pattern-based approach to modeling rules in section 3.5.

Object Constraint Language (OCL) is a formal language and included in UML as extension for defining constraints on a model [19, 21]. The main purpose of OCL is to specify restrictions on the possible system states with respect to a given model. OCL expressions are used to define invariants on classes and other types and also allow specification of pre- and post-conditions on operations. OCL does not have the capability to specify properties of the modeling language, or classify classes by means of meta classes and define their connectivity. Therefore, it is not applicable to our modeling problem.

2.2 Representation

Traditionally, different, often domain- or vendor-specific formats have been employed to represent conceptual models. Since the advent of the UML [19], UML is used by many

authors as the graphical notation for their models. Still, the question needs to be answered how to store the meta-models and domain models persistently. This need is emphasized in our environment, where we want to exchange the models between different tools. In such a setting, there are several requirements on the persistent representation of the models:

- The representation should be widely accepted by industry and standardized.
- The representation should be open, platform- and vendor-independent.
- The representation should be rich enough to encode all the required structures.
- The representation should support the interoperability and integration among applications and systems.
- The models described by the representation should be easy to verify.

There are many potential candidates that satisfy the requirements mentioned above, especially the different variants and developments around XML: the XML Metadata Interchange Format (XMI) [19] has been created by the OMG mainly for easy exchange of metadata between UML-based modeling tools and between MOF-based metadata repositories. MOF (Meta Objects Facility) [18] defines a meta-modeling framework that consists of four layers: data, model, metamodel and metametamodel (or MOF model). MOF provides three main metadata-modeling constructs: class, association and package. It is a standard modeling framework proposed by OMG and also widely used in the industry. UML (Unified Modeling Language) [19] is an important industrial standard language for object-oriented modeling, analysis and design. XMI integrates the power of UML in object-oriented modeling, analysis and design, the standard repository infrastructure of the MOF and the benefits of XML in the description of meta-data. It covers the transfer of UML and MOF-based data. It provides XML generation rules for producing XML metadata or documents from the corresponding MOF models. The OMG has already specified a UML DTD and a MOF DTD for representing UML models and MOF metadata. Unfortunately, the UML DTD and the MOF DTD are DTDs (Document Type Definitions), which have very limited modeling concepts and do not follow the syntax of XML and are thus treated differently from XML data.

XML schema [23, 3] is an alternative to using DTDs and is already recommended by the W3C. XML schema is a more powerful meta-data definition language. The basic language primitives are ComplexType, element and attribute. It supports different kinds of inheritance mechanisms such as extension and restriction, various data types, complex structures, and structure constraints. It provides substitution mechanisms, forced substitution (abstract) and data reuse on the document level. It is a good candidate for representing UML and MOF. But currently there is no XML schema for XMI.

RDF (Resource Description Framework) [15, 5] is designed for interoperability between applications that exchange machine-understandable information via the Web. RDF provides a mechanism for processing metadata, which could be interesting in our context. However, RDF provides only relatively limited modeling primitives such as object identify, binary relationships, and different containers (bag, sequence, alternative). It does not define an inheritance mechanism, content model, substitution, abstract elements or types etc. as does XML schema. This could prevent it from becoming a standard metadata language in the industry.

3 A Meta-modeling Approach

In this section we describe the different levels of a model from the process industry. The model discussed is only a small subset of what is needed to describe an actual plant, but it exemplifies the concepts needed and accurately follows the intention of the different levels.

3.1 The Modeling Problem

Technical systems usually consist of a number of different components, which are products of certain types. For example, a petrochemical process consists of a system of pumps, valves, tanks, pipes, etc. A company works on projects, which are related to actual plants, consisting of instances of these components, e.g. a plant P1 may contain a pump P01, a valve V01 and a tank F01, etc.

There is however usually not one view of the plant, which proves useful for all kinds of applications, but each application has different aspects, it is interested in. A plant can be seen from a location view, indicating the locations of the different elements of the equipment. A functional view describes the information exchanges between the different components on the level of abstraction suitable for a given application. Coming back to our example of petrochemical processes, the functional view might describe the equipment of the plant in terms of processes (mutable properties of the equipment involved in the chemical process), while the location view may be interested in the facilities (basic immutable properties of equipment).

The different views and models are used by different types of applications, like process databases, graphical online displays, simulators, asset optimization applications, engineering tools, diagnosis applications, and the like. To capture the required information the model should consist of the following three layers:

- A meta model layer describing the modeling concepts to be used. This includes classes to model equipment types and their connections in a topology, different kinds of variables and constants, mechanisms to describe time, abstraction concepts to structure the model, etc.
- A domain model layer describing the different types of equipment. Each type of equipment is characterized using the concepts from the meta level. A process for example is described by its variables and equations. On the model level, variants for the equipment type (e.g. equipment from different vendors) have to be modeled, e.g. using inheritance.
- A project data layer describing the plant under consideration. It consists of objects, which are instances of the domain model classes. These instances describe actual equipment installed in the plant. The number of objects on this level can be huge.

Meta information on the model is needed for different purposes:

- The applications need meta information to detect the domain classes, they are interested in. For example a display needs to detect, which values should be displayed in which format. An iterative simulator must find the source nodes with which to start the simulation.

- The meta information structures the model, so that the different views can be based on one common, consistent information model. An example for this is the modeling of petrochemical equipment by means of facilities and processes.
- If the structures described by the meta model are assumed by the the different applications, it must be ensured that engineers actually utilize these structures. Thus, the meta model provides modeling rules to be followed, when creating an application. The meta model must be explicitly represented, so that the domain model and project data can be checked for consistency. In this way the modeling rules are explicitly enforced.

3.2 An Example

The example captures all levels of a very simple structural model, describing the connections among a pump, a valve, and a tank. Fig. 1 shows a simplified meta model for this purpose. The meta class *Process* is the basis¹. All processes will be modeled as instances of this meta class. Composite processes are defined by aggregation of simple processes.

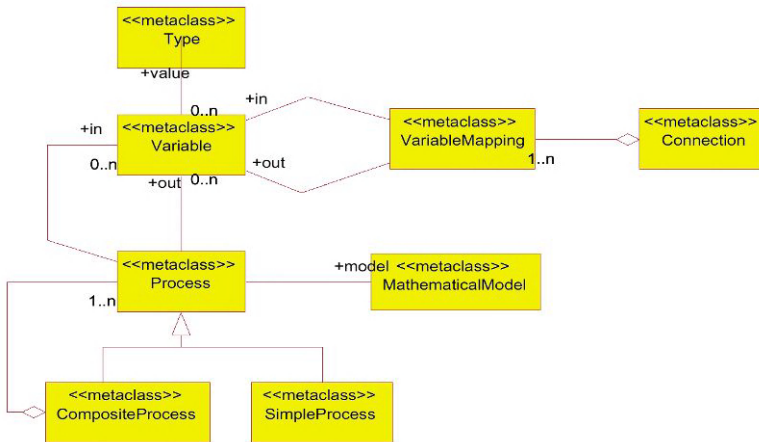


Fig. 1. Meta Model capturing Process Modeling Rules

A process has variables going into the process and variables coming out of the process. Variables have values, which belong to a certain variable *Type*. Processes are logically connected through a concept called *Connection*. A *Connection* consists of many variable mappings. A *VariableMapping* logically connects two variables.

¹ In a real-world scenario we would probably like to model that a piece of equipment can be used for multiple different processes, however, for the sake of simplicity we identify here the process (e.g. "Pumping") with the piece of equipment (e.g. "Pump") used for it.

Let us now look at a few diagrams from a domain model instantiating this meta model. The first part of the example shown in Fig. 2 describes a *Valve*. In this very simple version, a valve has 7 variables, each of which are modeled as instances of the meta class *Variable*. Four of these variables are input variables, as can be seen from the associations: the variables *FlowIn*, *TemperatureIn*, *PressureIn*, and *Position* are connected to the valve via associations, which are instances of the association *InputVariable*. The instance relationships among the associations is indicated by a dashed arrow, which indicates instance or dependency in UML. In addition, there are three output variables, *FlowOut*, *TemperatureOut* and *PressureOut*, which are connected to the valve via instances of the meta association *OutputVariable*. Further, valve has a *ValveModel*, which is an instance of *MathematicalModel*. Similar models are used for the pump and the tank.

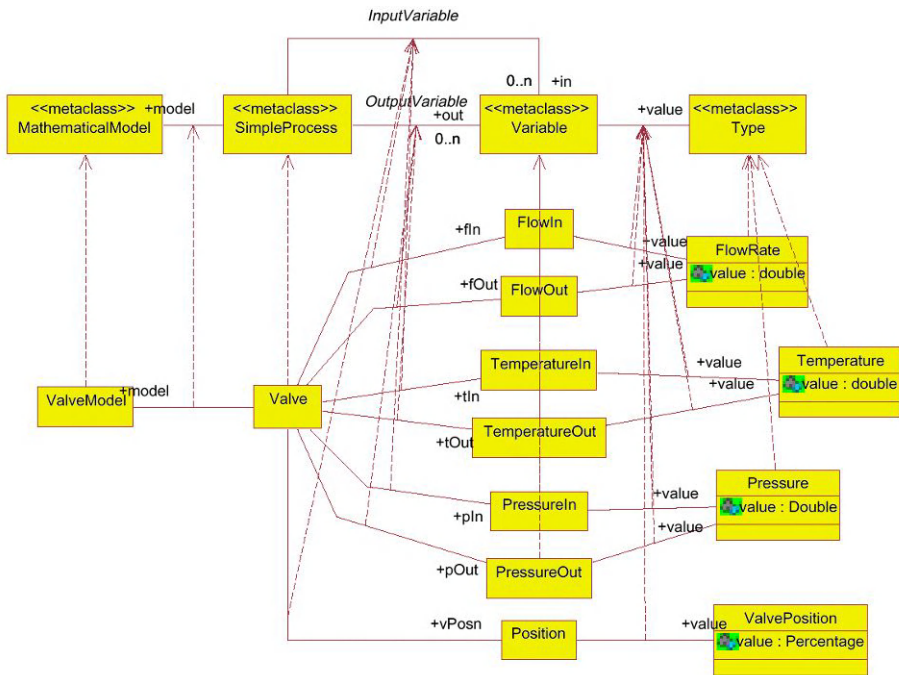


Fig. 2. Simplified Connectivity Model for a Valve

Once we have defined the simple processes, like pump, valve and tank, we connect them to model the topology of the plant. A connection class is shown in Fig. 3. As specified in the meta model, the *Connection* class *SimpleConnection* is an aggregation of several *VariableMapping* classes: *FlowMapping*, *PressureMapping*, and *TemperatureMapping*. Each of these classes connects two variables. For example, a *FlowMapping* connects the variable *FlowOut* (e.g. of a Pump) to *FlowIn* (e.g. of a Valve).

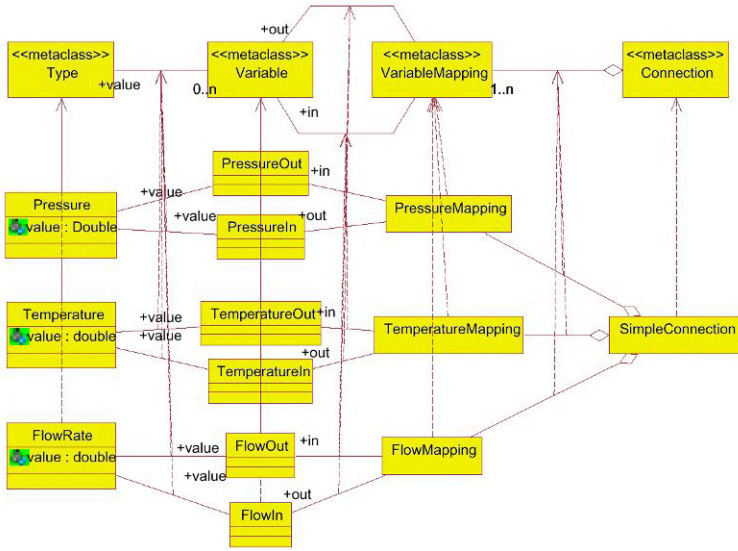


Fig. 3. A Connection class suitable for connecting Pump, Valve and Tank

3.3 Mathematical Representation

Now let us map the diagrams for meta level and model level to mathematical structures. A diagram is encoded by a structure

$$D = \langle CN, DCN, MCN, A, Agg, Mult, \sqsubseteq, \in_{CN}, \in_A \rangle$$

CN is a set of class names. CN consists of two disjoint partitions: domain classes DCN and meta classes MCN , i.e. $CN = DCN \cup MCN$.

A is a set of association names. A set $Agg \subseteq A$ denotes the set of all aggregation associations. For an association $a \in A$, $Classes(a)$ denotes the classes at the ends of a , i.e. $Classes : A \rightarrow (CN \times CN)$. For an association $a \in A$, $Mult(a)$ denotes the multiplicity of the association ends (for each of the ends, the multiplicity is given as a subset of the natural numbers): $Mult : A \rightarrow P(N) \times P(N)$.

Two types of relationships among classes are denoted separately. Generalization is modeled as a relation $\sqsubseteq \subseteq CN \times CN$. There are two relationships encoding instantiation, for classes $\in_{CN} \subseteq DCN \times MCN$ and for associations $\in_A \subseteq A \times A$.

As an example for this mathematical representation, consider Fig. 4. The diagram is formalized as follows:

$$\begin{aligned} MCN &= \{Process, CompositeProcess, SimpleProcess, Variable\} \\ DCN &= \{Source1, O1, O2\} \\ CN &= MCN \cup DCN \\ A &= \{Agg1, InputVariable, OutputVariable, AO1, AO2\} \\ Agg &= \{Agg1\} \\ \sqsubseteq &= \{(CompositeProcess, Process), (SimpleProcess, Process)\} \end{aligned}$$

$$\in_{CN} = \{(Source1, SimpleProcess), (O1, Variable), (O2, Variable)\}$$

$$\in_A = \{(AO1, Out putVariable), (AO2, Out putVariable)\}$$

$$Classes(Agg1) = (CompositeProcess, Process)$$

$$Classes(InputVariable) = (Process, Variable)$$

$$\vdots$$

$$Mult(Agg1) = (\{1\}, N \setminus \{0\})$$

$$Mult(InputVariable) = (\{1\}, N)$$

$$\vdots$$

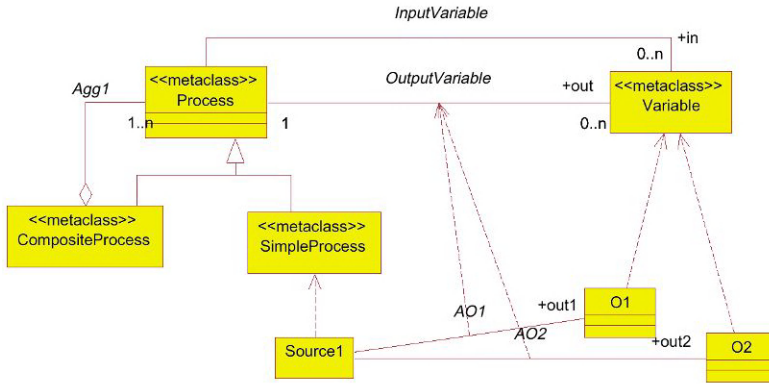


Fig. 4. Restricted example for demonstrating Representations

3.4 Defining Consistency

Having this mathematical representation in place, we can now define, when a domain model is consistent with a meta model. For this definition, we need the concept of a subclass.

Definition 1 ($Sub(c)$) For classes $c, d \in CN$, we write $c \sqsubseteq^+ d$ to denote that c is a direct or indirect subclass of d , i.e.

$$c \sqsubseteq^+ d \Leftrightarrow (\exists n \geq 0, \exists d_1, \exists d_n \in CN : c \sqsubseteq d_1 \sqsubseteq \dots \sqsubseteq d_n \sqsubseteq d)$$

For a class $c \in CN$, $Sub(c)$ denotes the set of all direct or indirect subclasses of c , $Sub(c) = \{b : b \sqsubseteq^+ c\}$.

The following constraints define, how to correctly instantiate a meta model. First, each domain class may be an instance of at most one meta class:

Constraint 2 (Unique Meta Class)

For each $d \in DCN$, if there exist metaclasses $m_1, m_2 \in MCN$ so that $d \in_{CN} m_1$ and $d \in_{CN} m_2$, then $m_1 = m_2$.

We do not restrict the number of instances of a meta class, nor do we postulate, that every class is an instance of a meta class. However, the associations in the meta model may only be instantiated between classes, which are instances of the correct meta classes.

Constraint 3 (Correctness of meta association instances)

For an association $a \in A$, so that $Classes(a) = (c_1, c_2)$ and $a \in_A a'$, where $a' \in A$ is a meta association so that $Classes(a') = (m_1, m_2)$, $m_1, m_2 \in MCN$, the following conditions must hold:

- c_1 must be an instance of m_1 or one of its subclasses, i.e. there has to be a class $m \in \{m_1\} \cup Sub(m_1)$, so that $c_1 \in_{CN} m$.
- c_2 must be an instance of m_2 or one of its subclasses, i.e. there has to be a class $m' \in \{m_2\} \cup Sub(m_2)$, so that $c_2 \in_{CN} m'$.

As an example for this constraint, consider again Fig. 4. *AO1* is an instance of *OutputVariable*, $Classes(AO1) = (Source1, O1)$, $Classes(OutputVariable) = (Process, Variable)$. According to the constraint, *Source1* must be a instance of *Process* or a subclass, which is true, since it is an instance of *SimpleProcess*, which is a subclass of *Process*. Further, *O1* must be an instance of *Variable* or a subclass, which is also true, since it is a direct instance of *Variable*. Thus, the constraint ensures, that associations instances have the right classes at their ends. The following constraint defines the number of instances of a meta association.

Constraint 4 (Number of meta association instances)

Let $a' \in A$ be a meta association between meta classes $Classes(a') = \{m_1, m_2\}$, where $m_1, m_2 \in MCN$. Let $Mult(a') = (M_1, M_2)$

- For each $m \in \{m_1\} \cup Sub(m_1)$ and each $c_1 \in DCN$, so that $c_1 \in_{CN} m$, the number of instances of a' connecting c_1 to other domain classes is given by its multiplicity:

$$\{c_2 : \exists a \in A : a \in_A a', Classes(a) = (c_1, c_2)\} \models M_2$$

- For each $n \in \{m_2\} \cup Sub(m_2)$ and each $c_2 \in DCN$, so that $c_2 \in_{CN} n$, the number of instances of a' connecting c_2 to other domain classes is given by its (other) multiplicity.

$$\{c_1 : \exists a \in A : a \in_A a', Classes(a) = (c_1, c_2)\} \models M_1$$

This constraint can again be explained using the example in Fig. 4. Let us consider the meta association *OutputVariable*, and the first bullet of the constraint: It talks about the domain classes instantiating the first end of the association. *Source* is such a domain class. The item says the the number of *OutputVariable* instances starting at *Source* is in the multiplicity defined for the other end of *OutputVariable*. There are two instances of *OutputVariable* starting at *Source*; the multiplicity given is 0..n, i.e. N , so the number of instances is in the allowed range ($2 \in N$). Let us look at the second item in the constraint: It talks about the domain classes instantiating the second end of the association. *OI* is such a domain class. The number of instances starting at *OI* is 1, which is consistent with the multiplicity given ($1 \in \{1\}$).

The next constraint talks about inheritance in the domain model. Since the meta classes are the basic categories for the domain model classes, generalization is only allowed between classes, which belong to the same meta class:

Constraint 5 (Inheritance in the Domain Model) *Let $c_1, c_2 \in DCN$ be domain classes. If $c_1 \in Sub(c_2)$ or $c_2 \in Sub(c_1)$, then there must exist a meta class $m \in MCN$, so that $c_1 \in_{CN} m$ and $c_2 \in_{CN} m$.*

To conclude, we postulate the correct usage of UML. We use UML to denote the meta-model, the model and their relations. The UML diagrams we use must satisfy the basic consistency constraints of the UML, some of which are given in [7].

Constraint 6 (Basic UML Constraints)

- *Generalization is not circular: For a class $c \in CN$, c is never a subclass of itself, i.e. $c \notin Sub(c)$.*
- *Associations exist only within the meta level or within the domain model level, not between entities from different levels:*

$$A \subseteq (DCN \times DCN) \cup (MCN \times MCN)$$

Finally we define consistency by postulating that all these constraints are satisfied:

Definition 7 (Consistent Model) *A model*

$$D = \langle CN, DCN, MCN, A, Agg, Mult, \Xi, \in_{CN}, \in_A \rangle$$

is called consistent, if D satisfies the constraints defined above.

Section 4 discusses, how models are stored, and when consistency according to these constraints is checked. But first, we compare the instantiation concept just defined to pattern instantiation.

3.5 Comparison to Patterns

Our meta-modeling technique has the goal to specify structure constraints in conceptual models. Patterns are predefined structures, which can be instantiated in conceptual model-

ing. Due to their similar purpose, a comparison of the two formalisms is worthwhile. Patterns have become a popular mechanism for representing good elements of software designs since [11], and have been applied to conceptual models [8]. The process of pattern instantiation has been studied in [20].

Fig. 5 shows a composite pattern with a correct instance, as discussed in [20]. The idea is that *Line* and *Text* are instances of *leaf*, *Graphic* is an instance of *Component* and *Picture* is an instance of *Composite*. Each element (class, association) of the instance is a renaming of an element in the pattern.

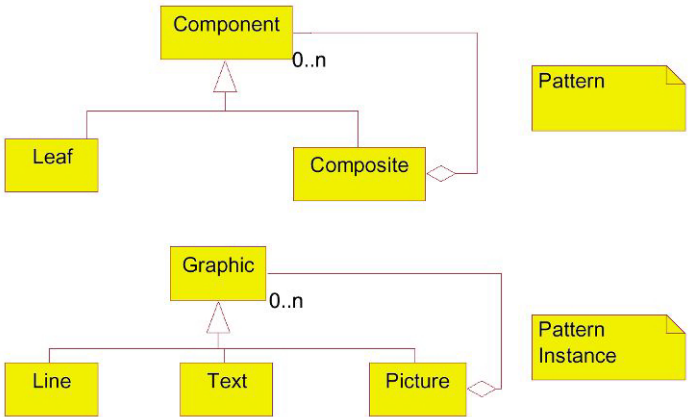


Fig. 5. Composite Pattern with Instance from [20]

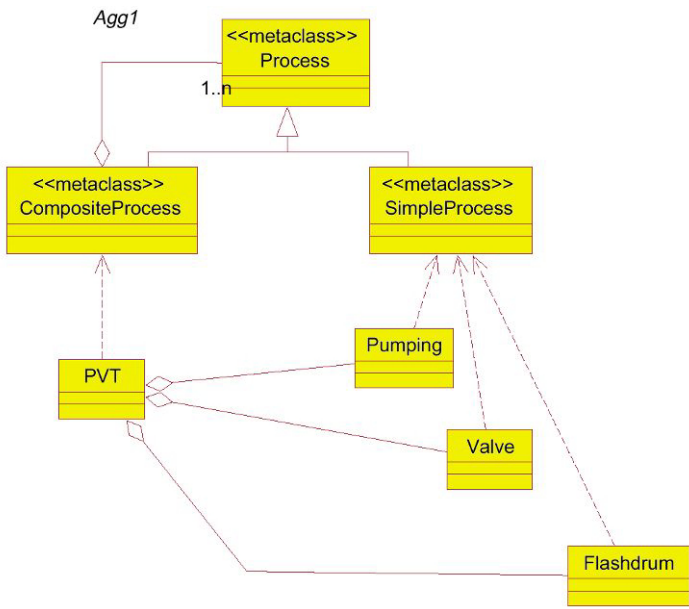


Fig. 6. Composite Process as Instance of Meta Model Classes

Obviously, the meaning of instance in the pattern world is different from a meta model instance. A pattern instance contains renamed copies of the pattern elements, while the meta model instance contains elements satisfying the specifications in the meta model. This can be highlighted with the association instances: The pattern instance contains renamed copies of the associations. The multiplicities of the instantiated associations are the same as in the pattern. In contrast to that, the meta associations define the number of allowed instances with their multiplicity (see Fig. 6). Since the multiplicity of *Process* is 1, n in the aggregation association of *CompositeProcess*, each instance of composite process may contain any number (in this case three) of instances of *Process*. A further difference is that the pattern consists of classes, not meta classes. This means, that the instantiation relationship between meta class and class is not explicitly represented in the model. Our approach represents the instantiation explicitly, so that correct instantiation can be checked automatically, and applications can query the model based on meta classes, e.g. they can ask for all processes defined in the system. Finally, when using patterns, a domain model contains pattern instances (among other elements). When using meta models, the domain model as a whole is an instance of the meta model. This example has shown, that patterns are templates for parts of conceptual models, i.e. they determine the structure of their instances, which are renamed copies. In contrast to that, meta models are specifications of conceptual models.

4 Persistent Representation and Information Exchange

Up to now we have defined a meta-modeling formalism with consistency conditions, a graphical UML representation and a formal mathematical representation.

It would be ideal to have a standard language with rich enough syntax to capture all parts of the model and rich enough semantics, so that a parser for the language could check our consistency conditions. Then, applications could simply use the standard parser to access, modify, update and exchange models according to our approach. Fig. 7 shows how such a language could represent our models. XMI has some potential to provide at least the syntactic richness for such an approach in the future. However the current version is based on a DTD and shows some severe limitations (for example, when dealing with multiplicity constraints) and is not suitable for representation of our models. It is not clear, if in the future a common standard for expressing meta models will develop based on XML, and if such a standard will accidentally have the exact semantics required for our approach.

Since we have to live with today's languages and tools, we have chosen another approach for the representation of our models, which is depicted in Fig. 8. We use an XML schema to represent the syntax of our modeling language. This schema is very similar to the mathematical structure defined in section 3.3. All levels of the model, including the project data are stored in (one or more) XML document(s) according to the XML schema. A standard parser for XML schemata only partly verifies the consistency of the model. The parser verifies for example, that each class has a name, and that multiplicities are given. However, the XML parser cannot check, if the number of association instances is correct, or if the classes at the association ends are as specified.

These constraints are implemented by a custom consistency checker.

Next, to conclude our example, we list a (simplified) schema for our formalism and a document for the example shown in Fig. 4.

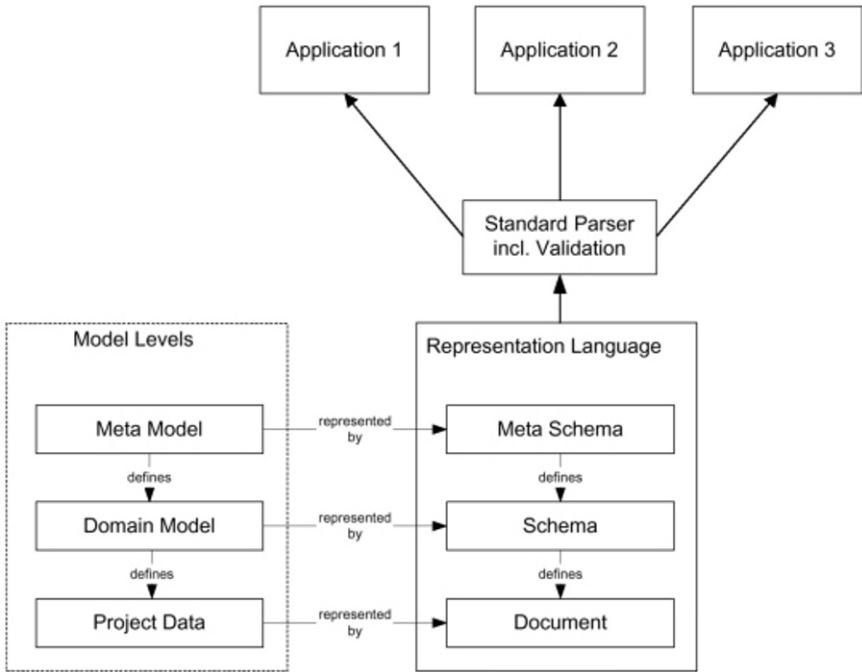


Fig. 7. Representation using a (currently not available) standard meta model representation

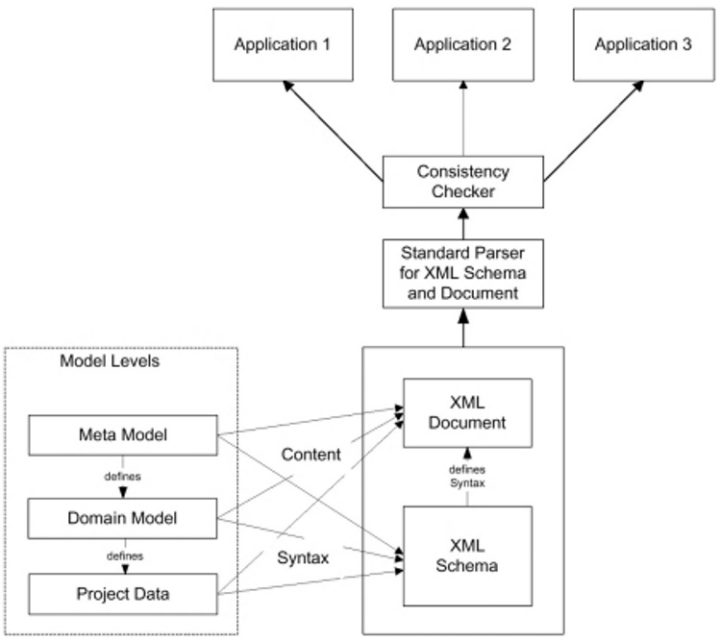


Fig. 8. Representation of the model using XML Schema

4.1 Schema

Definition of Classes

```
<xsd:element name="Class" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="metaclass" type="xsd:boolean" use="required" default="false"/>
  </xsd:complexType>
</xsd:element>
```

Definition of Associations

```
<xsd:element name="Association" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:sequence minOccurs="2" maxOccurs="2">
      <xsd:element name="AssociationEnd">
        <xsd:complexType>
          <xsd:attribute name="className" type="xsd:string" use="required"/>
          <xsd:attribute name="roleName" type="xsd:string" use="optional"/>
          <xsd:attribute name="minOccurs" type="xsd:integer" use="required"/>
          <xsd:attribute name="maxOccurs" type="xsd:integer" use="required"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="aggregation" type="xsd:boolean" use="required"/>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
```

Definition of Generalisation

```
<xsd:element name="Generalisation" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:attribute name="super" type="xsd:string" use="required"/>
    <xsd:attribute name="sub" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
```

Definition of Instantiation

```
<xsd:element name="AssociationInstantiation" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:attribute name="concept" type="xsd:string" use="required"/>
    <xsd:attribute name="instance" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ClassInstantiation" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:attribute name="concept" type="xsd:string" use="required"/>
    <xsd:attribute name="instance" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
```

4.2 Document

```

<Class name="CompositeProcess" metaclass="true"/>
<Class name="SimpleProcess" metaclass="true"/>
<Class name="Variable" metaclass="true"/>
<Class name="Source1" metaclass="false"/>
<Class name="O1" metaclass="false"/>
<Class name="O2" metaclass="false"/>
<Association aggregation="true" name="Agg1">
  <AssociationEnd className="CompositeProcess" minOccurs="1" maxOccurs="1"/>
  <AssociationEnd className="Process" minOccurs="1" maxOccurs="-1"/>
</Association>
<Association aggregation="false" name="InputVariable">
  <AssociationEnd className="Process" minOccurs="1" maxOccurs="1"/>
  <AssociationEnd className="Variable" minOccurs="0" maxOccurs="-1" roleName="in"/>
</Association>
<Association aggregation="false" name="OutputVariable">
  <AssociationEnd className="Process" minOccurs="1" maxOccurs="1"/>
  <AssociationEnd className="Variable" minOccurs="0" maxOccurs="-1" roleName="out"/>
</Association>
<Association aggregation="false" name="AO1">
  <AssociationEnd className="Source1" minOccurs="1" maxOccurs="1"/>
  <AssociationEnd className="O1" minOccurs="1" maxOccurs="1" roleName="out1"/>
</Association>
<Association aggregation="false" name="AO2">
  <AssociationEnd className="Source1" minOccurs="1" maxOccurs="1"/>
  <AssociationEnd className="O2" minOccurs="1" maxOccurs="1" roleName="out2"/>
</Association>
<Generalisation super="Process" sub="CompositeProcess"/>
<Generalisation super="Process" sub="SimpleProcess"/>
<AssociationInstantiation concept="OutputVariable" instance="AO1"/>
<AssociationInstantiation concept="OutputVariable" instance="AO2"/>
<ClassInstantiation concept="SimpleProcess" instance="Source1"/>
<ClassInstantiation concept="Variable" instance="O1"/>
<ClassInstantiation concept="Variable" instance="O2"/>

```

5 Conclusion

In this paper we have shown how to apply meta-modeling to represent modeling rules for industrial applications. Such modeling rules are particularly useful, when information models have to be shared among different applications, each of which has its own requirements on the representation.

A mathematical representation of the meta model together with the domain model allows us to define precisely, when we consider a domain model a correct instance of the meta model. The different modeling levels are represented graphically using the UML notation and made persistent via an XML schema, which encodes the mathematical representation of model. The following can be observed:

- Requirements on the structure of the domain model are represented explicitly and checked automatically based on our definition of consistency. In this way our approach enforces modeling rules.
- Since the meta model is explicitly represented by means of meta classes and meta associations, it can be changed and extended (in contrast to hard-coded modeling rules represented by application code or constraints on specific domain model classes).
- The formalism is more adequate for our purpose than analysis patterns.

A prototype implementing these concepts has been developed and used on an example from the process industry. A productization project, which builds on these ideas is underway. Besides the implementation of our concepts we are further formalizing the foundations of our approach. It is part of our future work to consider the consistency definition more formally: In the current paper, we have defined, when we regard a model (domain model plus meta model) as consistent. The next step is to show, that a model which we consider consistent is always consistent in the mathematical sense [12], i.e. we have to formally define a semantics for our diagrams and show that each consistent diagram has a model (in the logical sense of the term).

References

1. Rafael Batres. Modelica as a model representation language of behavior models in MDF. Technical Report TR-JSPS-2000-06-22, Tokyo Institute of Technology, Japan, June 2000.
2. B. Bayer, R. Schneider, and W. Marquardt. Integration of data models for process design - first steps and experiences. In *7th International Symposium on Process Systems Engineering*, Keystone, Colorado, July 2000.
3. Paul V. Biron and Ashok Malhotra. Xml schema part 2: Datatypes. Technical report, World Wide Web Consortium, May 2001.
4. Ruth Breu, Radu Grosu, Franz Huber, Bernhard Rumpe, and Wolfgang Schwerin. Towards a precise semantics for object-oriented modeling techniques. In Haim Kilov and Bernhard Rumpe, editors, *Proceedings ECOOP'97 Workshop on Precise Semantics for Object- Oriented Modeling Techniques*. TUM-I9725, 1997.
5. Dan Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification 1.0 - W3C Candidate Recommendation. Technical report, World Wide Web Consortium, March 2000.
6. Rational Software Corporation et al. Uml semantics, version 1.1. Technical report, OMG, September 1997.
7. Andy S. Evans. Reasoning with UML class diagrams. In *Second IEEE Workshop on Industrial Strength Formal Specification Techniques, WIFT'98*, Boca Raton/FL, USA. IEEE CS Press, 1998.
8. Martin Fowler. *Analysis Patterns: Reusable Object Models*. Object Technology Series. Addison-Wesley, 1997.
9. Peter Fröhlich, Nicola Henze, and Wolfgang Nejdl. Meta-modeling for hypermedia design. In *Proc. of Second IEEE Metadata Conference*, Maryland, September 1997.
10. Peter Fröhlich and Wolfgang Nejdl. Webrc: Configuration management for a cooperation tool. In *Proceedings of the Seventh Workshop on Software Configuration Management (SCM'7)* (Springer LNCS 1235), Boston, 1997.
11. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley, 1995.
12. David Harel and Bernhard Rumpe. Modeling languages: Syntax, semantics and all that stuff - part I: The basic stuff. Technical Report MCS00-16, Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, Israel, September 2000.
13. M. Jarke, R. Galletseder, M. Jeusfeld, M. Staudt, and S. Eherer. Conceptbase - a deductive object base for meta data management. *Journal on Intelligent Information Systems*, 4(2): 167– 192, 1995.

14. S. Kent, A. Evans, and B. Rumpe. Uml semantics faq. In A. Moreira and S. Demeyer, editors, *Object-Oriented Technology, ECOOP'99 Workshop Reader*. LNCS 1743, Springer Verlag, 1999.
15. Ora Lassila and Ralph R. Swick. Ressource Description Framework (RDF) Model and Syntax Specification. Technical report, World Wide Web Consortium, February 1999.
16. M. L. Lu, R. Batres, H. S. Li, and Y. Naka. A G2-Based MDOOM Testbed for Concurrent Process Engineering. *Comput. Chem. Engng.*, 21, 1997. Supplement pp. S11–S16.
17. J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: A language for representing knowledge about information systems. *ACM Transactions on Information Systems*, 8(4), 1990.
18. OMG. Meta Object Facility (MOF) Specification Version 1.3 RTF. Technical report, Object Management Group Inc., September 1999.
19. OMG. OMG Unified Modeling Language Specification Version 1.3. Technical report, Object Management Group Inc., March 2000.
20. Bernd-Uwe Pagel and Mario Winter. Towards Pattern-Based Tools. In *Proceedings Euro- PLoP '96*, 1996.
21. Mark Richters and Martin Gogolla. On Formalizing the UML Object Constraint Language OCL. In Tok-Wang Ling, Sudha Ram, and Mong Li Lee, editors, *Proc. 17th Int. Conf. Conceptual Modeling (ER'98)*, pages 449–464. Springer, Berlin, LNCS 1507, 1998.
22. W.N. Robinson and S. Volkov. A Meta-Model for Restructuring Stakeholder Requirements. In *Proceedings of the 19th International Conference on Software Engineering (ICSE'97)*, Boston, USA, May 1997. IEEE Computer Society Press.
23. Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. Xml schema part 1: Structures. Technical report, World Wide Web Consortium, May 2001.

Modelling Ubiquitous Web Applications – The WUML Approach

Gerti Kappel¹, B. Pröll², Werner Retschitzegger¹, and Wieland Schwinger³

¹ Institute of Applied Computer Science, Department of Information Systems (IFS)
University of Linz, Altenbergerstraße 69, A-4040 Linz, Austria
{gerti | werner}@ifs.uni-linz.ac.at

² Institute for Applied Knowledge Processing (FAW)
Softwarepark Hagenberg, Hauptstraße 99, A-4232 Hagenberg, Austria
bproell@faw.uni-linz.ac.at

³ Software Competence Center Hagenberg (SCCH)
Softwarepark Hagenberg, Hauptstraße 99, A-4232 Hagenberg, Austria
wieland.schwinger@scch.at

Abstract. E-commerce and m-commerce have dramatically boosted the demand for services which enable ubiquitous access. Ubiquity with its anytime/anywhere/anymedia nature requiring context-aware computing calls for new engineering techniques supporting these kind of services. In this paper, we propose the notion of customisation as the uniform mechanism to deliver ubiquitous web applications providing adaptability with respect to a certain context. As a prerequisite for supporting customisation design, a set of generic models is introduced comprising a context model, a profile model, and a rule model. At the application's side, customisation hooks are provided representing the major hot spots of adaptation. A customisation toolkit in terms of a customisation rule editor and browser supports an integrated modelling process and facilitates reusability on the basis of a repository of customisation rules and patterns.

1 Introduction

The Internet and in particular the World Wide Web have introduced a new era of computing, providing the basis for promising application areas like e-commerce and m-commerce [10], [19]. These application areas have dramatically boosted the demand for services which enable *ubiquitous access*, being mainly characterised by the *anytime/anywhere/anymedia paradigm* [23]. Considering ubiquitous web applications from a software engineering point of view, as their complexity increases, so does the importance of modelling techniques [1], [3], [7]. Models of a ubiquitous web application prior to its construction are essential for comprehension in its entirety, for communication among project teams, and to assure architectural soundness and maintainability. There are already a couple of methods especially dedicated to the modelling of web applications, which however lack proper support for the issues arising when dealing with ubiquity [12].

We propose that a ubiquitous web application should be designed from the start taking into account not only its hypermedia nature, but also the fact that it must run "as is" on a variety of platforms, including mobile phones, Personal Digital Assistants (PDAs), full-fledged desktop computers, and so on. This implies that a ubiquitous

web application has to take into account the different capabilities of devices comprising display size, local storage size, method of input and computing speed as well as network capacity. New opportunities are offered in terms of location-based, time-based, and personalised services taking into account the needs and preferences of particular users. Consequently, a ubiquitous web application must be *context-aware*, i.e., aware of the environment it is running in.

This paper establishes the notion of *customisation* as uniform mechanism to provide adaptability of a ubiquitous web application with respect to a certain context. In particular, the contribution can be summarised as follows:

- We give an overview on the state of the art of *ubiquitous web application engineering* and highlight the *different facets of customisation* from a historical point of view.
- We give insight into appropriate *customisation modelling concepts and notations* by proposing a *generic customisation model* which comprises a *context model*, a *profile model* and a *rule model* and by separating the application into a *stable part* and a *variable part* which provides appropriate *customisation hooks*.
- We outline first ideas about proper tool support for customisation design in terms of a *customisation toolkit* which provides a *customisation rule designer* and *browser* and we propose the abstraction of concrete customisation rules into *customisation rule patterns* which are part of a *pattern library* to facilitate reusability.

2 Customisation – The Notion and the Issues

Considering the notion of customisation from a historical point of view, it represents a major challenge at least since the end user has been put in the middle of concern when developing interactive applications. An area dealing with customisation issues already for a long time is the user interface community, which brought up the notion of *adaptive user interfaces*, cf., e.g., [9]. Adaptive user interfaces are designed to tailor a system's interactive behaviour considering both individual needs of human users and changing conditions within an application environment. The broader approach of *intelligent or advisory user interfaces* includes adaptive characteristics as a major source of its intelligent behaviour, cf., e.g., [5]. Another area dealing with customisation but emphasising more on adapting the content of an application are *information filtering and recommender systems* [2]. The goal of these systems is to go through large volumes of dynamically generated textual information and present to the user those which are likely to satisfy his/her information requirements. With the emerge of hypertext the need for alternative access paths to information in terms of, e.g., different navigation structures became prevalent leading to another research direction called *adaptive hypertext and hypermedia* [4]. Last but not least, the proliferation of *mobile computing* and *mobile web applications*, in particular, makes it necessary to consider not only the user preferences but also the environment in terms of, e.g., location information or device characteristics in order to adapt the application properly [17].

Customisation Dimensions. Learning from these different areas, the *design space of customisation* can be characterised by three orthogonal dimensions, comprising the *degree of customisability*, the *granularity of adaptation* and the *kind of context* (cf. Fig. 1) [15].

Degree of Customisability. The first dimension, the *degree of customisability*, expresses that both, context and adaptation can be either *static*, i.e., pre-defined or *dynamic*, i.e., determined during run-time. An example for static context and adaptation could be to select a pre-defined version of a certain application depending on the device used. An example for the fully dynamic case would be to adapt the resolution of an image on the fly, due to a change in bandwidth. Applications supporting only static contexts and/or adaptations are often called *adaptable* whereas those providing also dynamic concepts are considered to be *adaptive* [16].

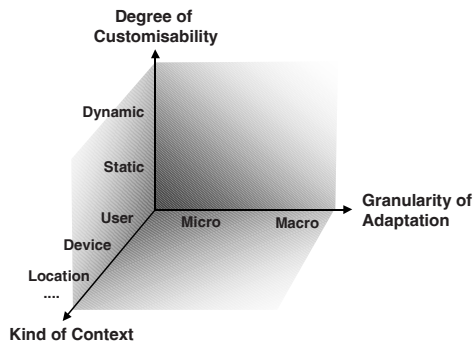


Fig. 1. Dimensions of Customisation

Granularity of Adaptation. The second design dimension indicates the *granularity of adaptation* ranging from *micro adaptation* to *macro adaptation*. Whereas micro customisation is concerned with fine-grained adaptations (e.g., disabling a link on a page), macro adaptation means that depending on the context, rather large parts of an application are adapted (e.g., instead of an indexed guided tour, use a simple bullet list). Note that, there is no exact border between micro and macro adaptation. In its most extreme form, macro adaptation simply means that depending on the context, the whole application realising a certain service is substituted by another one, thus better fitting in the current context. This extreme form of macro customisation often occurs together with the combination of static context and adaptation. As encountered in [15], the few existing approaches dealing with modelling aspects of ubiquitous web applications adhere in large parts to this kind of adaptation, e.g., by modelling several so-called "site views" each one tailored to another context.

Kind of Context. The third dimension covers the *kind of context* which is considered by customisation. In this respect, it is useful, to consider existing approaches to customisation as done in [12]. The majority of these approaches consider the issue of *personalisation* by making assumptions about relevant user characteristics and preferences to get personalised services from a certain resource, or even to personalise already the discovery of resources. Also a considerable number of approaches take *network and device properties* into account. Those two are often considered together which is reasonable since mobile devices also imply a wireless connection carrying certain network constraints. In [12] it has been encountered, however, that only few of the surveyed approaches explicitly regard location information. This is due to both technical deficiencies and lack of legal regulations.

Semantic Enhancement vs. Semantic Equivalence. Those approaches supporting personalisation or providing location information endow the application with *semantic enhancement*, in that each particular user is provided with specific added value. Such enhancement actually makes the same application provide increased value for different users, who ultimately perceive the application as two different services. On the other hand, the same application customised for the same user may (and certainly does) look different when it is run on different devices and/or in different situations. This is inevitable (for example it is impossible to show that beautiful applet on a PDA with no virtual machine installed), but the service (or the added value) provided to the user should nevertheless be the same. In this case customisation enables to maintain *semantic equivalence*. In this case we talk of semantic equivalence, which means that, despite the different context, the value provided to the user should still be the same.

3 The WUML Approach to Customisation Design

WUML stands for *Web Unified Modeling Language* and aims at the methodological support of web application development with special focus on ubiquity. The design goals behind WUML are

- usage of *UML* [22] as the basic formalism,
- web-tailored extension of UML using the *UML extension mechanisms* only,
- specification of WUML in terms of a *generic framework*,
- *three-level view* on the development process (cf. [21]) taking *content* (i.e., domain-dependent data) *hyperbase* (i.e. the logical composition of web pages together with the navigation structure), and *presentation* (i.e., the layout of each page as well as user interaction) into account, and
- support of *ubiquity* through *customisation*.

In this paper, we focus on the customisation model of WUML, for all other aspects we refer to [14].

Our approach to customisation design is based on a broad view on customisation in that it takes into account all three dimensions of customisation as identified in the previous section. First, although most often separated in existing approaches, we think that customisation for ubiquitous web applications should uniformly consider *personalisation* aspects, together with issues resulting from the *anywhere/anytime/anymedia paradigm*, herewith providing both, semantic enhancement and semantic equivalence. Second, customisation design should not only consider *macro adaptation* but also *more fine-grained forms of adaptation* in order to be able to better reflect a certain context when tailoring an application (again with respect to semantic enhancement and semantic equivalence). Third, since in our opinion, static context and static adaptation does not reflect the dynamic nature of ubiquitous web applications, a special focus of customisation design is given on dynamic context and dynamic adaptation.

Overall Architecture. The architecture given in Fig. 2 provides the basis for accomplishing such a broad view of customisation. The *context* together with *profiles* provide detailed information about the environment of an application and trigger the actual customisation as soon as the environment changes. Whereas context represents current and historical information about the environment of the application which is automatically monitored, profiles cover more stable information which are explicitly given by a designer or a user, e.g., user preferences or the vendor of a device, e.g.,

descriptions of device properties. A rule-based mechanism in terms of *customisation rules* is employed in order to specify the actual customisation. Customisation rules monitor changes in context, profile and application state itself. For separation of concerns, the application is separated into a *stable part*, comprising the default, i.e., context-independent structure and behaviour of the application and a *variable, context-dependent part*, thus being subject to most of the adaptations.

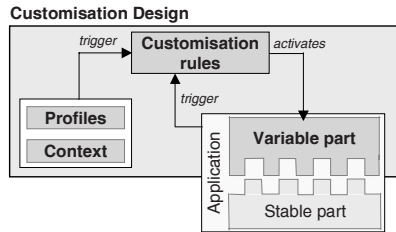


Fig. 2. Overall Architecture of the WUML Approach to Customisation Design

Customisation Design vs. Traditional Application Design. Finally, it has to be noted that customisation could also be made part of traditional application design without explicitly considering customisation rules, profiles, context and the variable part of an application separately from its stable part. By means of factoring out these aspects from the application already during the early stages of the software lifecycle, however, the dynamic nature of ubiquitous web applications can be much better dealt with, not least with respect to reusability and locality of changes. The rationale behind an explicit customisation design is similar to the motivation given for factoring out business policies from object-oriented applications [13].

4 Customisation Model

To support the architecture introduced in the previous section, we propose a *generic customisation model* in the sense of an object-oriented framework. Generic means in this respect that the model is domain-independent and provides a couple of pre-defined classes which can be used for customising a certain web application. In addition, the pre-defined classes can be extended by the designer through subclassing in order to cope with application specific details. The provided generic models are discussed in more detail in the forthcoming subsections.

4.1 Context Model

As already mentioned, we define context as the *reification of the environment in terms of its properties* which are continuously, dynamically updated to take into account the fact that the environment also continuously changes. For us the context is *not modifiable* and so is outside of the control of the ubiquitous web application. It just provides a manageable description of the environment so that it can be addressed within customisation design. Although the context is *application independent*, the designer needs to specify which properties of the environment are of interest for a particular application.

Context Properties. Based on the existing literature discussed in the previous section and considering what is currently made available by existing technology, our generic context model considers the following *context properties* which can be, as already mentioned, extended by the designer:

- *User Agent:* The user agent property refers to the demand of ubiquitous web applications for anymedia. It provides information about the device and browser. Together with a device profile and a browser profile (cf. Section 4.2) this allows to identify constraints relevant for a proper adaptation of the application.
- *User:* The knowledge about the user takes into account the necessity of personalisation. Looking at existing technology, the provided telephone number together with user profile information (cf. Section 4.2) allows identifying the individual user and user class.
- *Network:* Context properties concerning the network comprise, e.g., the bandwidth.
- *Location:* Location copes with the need for mobile computing and captures information about the location an application is accessed. Actually, this information is not directly provided by mobile devices themselves but is obtained via a so-called location server.
- *Time:* Finally, the context property time allows to customise the application with respect to certain timing constraints such as opening hours of shops or timetables of public transportation. Currently the time context property is represented at the server only.

Session, Current Context and History. Since web applications enforce the notion of sessions, possibly consisting of a sequence of transactions, these context properties need to be considered within the boundaries of sessions, i.e., each session has its own context. Furthermore, since the context within a session is subject to continuous changes, it is necessary to identify the most recent reification of the environment, which is further on called *current context*, using the latest timestamp. The current context comprises the current values of the context properties for a given interaction (e.g., the most recent) within the session of a ubiquitous web application. Practice has shown that it is necessary to broaden the view on context in that a context does not only consider the current context at a given point in time but also historical information. This is necessary to be able to identify changes in the values of the context properties over time. Thus the context model also needs to include a *history dimension*, in that a relevant context C can be formulated as a vector of context property values over time. For example, to determine user navigation patterns or the average throughput of a system, it is necessary to collect information about historic interactions. In contrast, the information about which device is used allowing customising the presentation to fit to a restricted display size mostly requires information of the current device only.

Our understanding of an appropriate context model is shown in Fig. 3 by means of a UML class diagram. Since every context information needs to be considered within a session the class *Session* acts as its root and at the same time holds the collection of all session instances within the system. For each session the class *History* can hold a sequence of contexts indexed by timestamps. The context, represented by the class *Context*, is an aggregation of a number of context properties, each of which is a subclass of the abstract *ContextProperty* class. Note that the *Context* class is not an aggregation of the generic abstract class but rather of the specific, concrete subclasses.

We chose this representation because we prefer to have a *context vector* that explicitly represents the specific subclasses herewith emphasising the fact that at a certain point in time only one instance of each specific context subclass may be present.

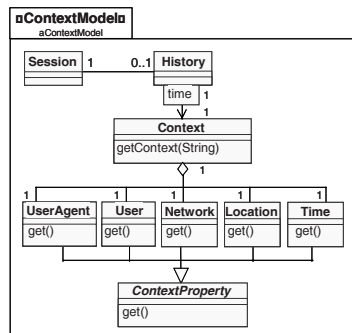


Fig. 3. Context Model

4.2 Profile Model

Profiles, in contrast to the context, represent more stable information which is explicitly given by, e.g., a user, a designer or the vendor of a device. The proper representation of profile information is already subject to standardisation efforts. The World Wide Web Consortium (W3C) is working on a framework for the management of device and user profile information called "*Composite Capabilities / Preference Profiles*" (CC/PP) which is based on the Resource Description Framework (RDF) [24]. It specifies how client devices express their capabilities and users express their preferences to the web application server. In addition, it defines recommendations about the content of such profiles. One major goal followed by CC/PP is extensibility so that new properties can be defined and included in the description and users can overwrite vendor-defined default preferences.

We build on this standardisation effort when modelling profile information. Depending on the kind of profile, information may be available already at *design time*, e.g. device characteristics, or, if the profile is application dependent, not before *run-time*, e.g. usage statistics. In the latter case, the designer of the ubiquitous web application needs to take care to explicitly provide the information throughout the application run-time. For this the profile information needs to be considered as any other information.

Analogous to the context model described in the previous section, our profile model encompasses the following profiles:

- *User Agent*: User agent profiles describe both, the capabilities of devices, e.g., display size, memory, operating system and the browsers running on these devices, e.g., kind of browser and version number. User agent profiles are application independent.

- *User*: A user profile can comprise information that is *voluntarily entered* by the user describing the user's preferences with respect to application customisation as well as information that is *transparently acquired* by the system including, e.g., usage statistics. Note that in general the user profile is application dependent as opposed to the user agent profile and the context.
- *Network*: A network profile is application independent and could, e.g., contain maximal bandwidths of certain connection types.
- *Location*: An example for a location profile is a road map which is application independent.
- *Time*: Finally, a time profile, which is again application independent, could encompass time zones or time-of-day settings.

Similar to our generic context model, the profile model is not comprehensive but can be easily extended by the designer to represent additional kinds of profiles necessary for a particular ubiquitous web application.

4.3 Rule Model

For specifying a certain customisation we propose the usage of rules (furtheron called *customisation rules*) in terms of the *event/condition/action (ECA) mechanism*. ECA rules are well known in the area of active database systems [11] and represent among others a commonly accepted mechanism for a localised specification of business policies as opposed to spreading it over several applications [13]. Although it would be possible to use CA-rules only, we stick to an event-driven specification to better reflect the dynamic nature of ubiquitous web applications. In our approach, the event together with the condition describes the *situation* when customisation is necessary.

Event. The event part of a customisation rule determines the events which are able to trigger the rule. With respect to our overall architecture given in Fig. 2, it detects changes in context, profile or application state and therefore indicates the need for a potential customisation. An example for an event would be a change of bandwidth. For more details it is referred to Section 4.4.

Condition. The condition part is evaluated as soon as the rule is triggered by an appropriate event and evaluates whether and which adaptation is actually desired. An example would be to test whether the bandwidth falls below a certain minimum. Conditions are in fact predicates which can be combined by means of logical operators currently using OCL-syntax [22].

Action. The action part of a customisation rule is responsible for activating a certain adaptation of the application, e.g., switch to text mode. An action mainly deals with so called *customisation hooks* as provided by the designer of the ubiquitous web application (cf. Section 5) but can also activate other operations of the stable part of the application which are not explicitly foreseen for customisation purposes.

Rule Properties. Several properties of a rule, such as *priority*, *activation state* and *transaction mode* [11], may be used to further specify the actual customisation process at runtime. This in turn determines when the events are detected, when the condition is evaluated, and when the action is executed.

4.4 Event Model

Applying the ECA mechanism in the domain of ubiquitous web applications requires a dedicated event model. The events of this event model need to monitor changes within both, context and profile information as well as changes in the application state, resulting from explicit user requests. Consequently, our event model considers three basic types of *primitive events* which in turn can form *composite events*. Fig. 4 shows a UML class diagram of our event model. Since we consider events as first-class objects, the application developer is able to extend the pre-defined event types by means of subclassing.

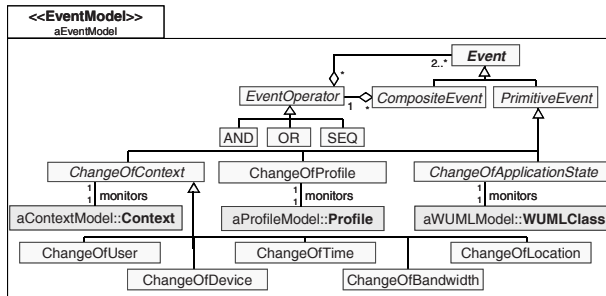


Fig. 4. Event Model

The event model provides the following three basic types of primitive events:

- *Change of context*: We propose the following pre-defined events indicating changes in the context, namely, `ChangeOfUser`, `ChangeOfDevice`, `ChangeOfBandwidth`, `ChangeOfLocation`, `ChangeOfTime`. Note, that these events directly monitor changes within the corresponding context properties of our context model.
- *Change of profile*: Since the properties of certain profiles are not necessarily limited, we propose at a first attempt a single pre-defined event only, namely `ChangeOfProfile`.
- *Change of application state*: Changes in the application state, resulting, e.g., from a certain user interaction may signal events, which are useful for customisation purposes. There exists a number of pre-defined event classes which are specific to the modelling elements of WUML and therefore not depicted in Fig. 4.

To be able to model complex real-world situations, which should be monitored, we suggest the notion of so-called *composite events*. Composite events are constructed by means of the logical event operators `AND`, `OR`, and `SEQ`, using the above mentioned primitive events or again composite events as operands. For example, the requirement that a page should be both personalised and tailored to a certain device demands for a composite event like `(ChangeOfUser OR ChangeOfDevice)`. Composite events also allow to express if the actual adaptation should be done in advance, i.e., independent of any user's request or on the fly, meaning that adaptation is not done before it is needed in response to a user's request. In the first case, customisation rules monitor changes in context, profile or application state only, whereas in the latter, the request of a user has to be taken into account too.

5 Customisation Notation

5.1 Notation for Specifying Customisation Hooks

As a major prerequisite for realising customisation we propose that the application should provide appropriate *customisation hooks*, herewith again resembling a basic idea of object-oriented frameworks. These customisation hooks provide the basis for customisation rules to adapt the application towards a particular context. As already mentioned, the application is separated into a *stable part* that provides the basic functionality, and a *variable part* that provides the customisation hooks for both, micro customisation and macro customisation. Whereas we consider micro customisation at a very fine-grained level, meaning that a *single modelling element is adapted* only, macro customisation which is based in turn on micro customisation (cf. below) covers the adaptation of *more than one modelling element*.

Micro Customisation. For modelling micro customisation, we propose that the UML model element class, representing a basic modelling element in WUML should be extended with two additional compartments for representing the variable part of the application in terms of customisation hooks (cf. Fig. 5).

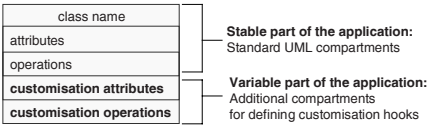


Fig. 5. Additional Class Compartments for Specifying Customisation Hooks

Macro Customisation. Concerning macro customisation, we propose to use the UML *package mechanism* in order to group those modelling elements which are subject to the same adaptation within so called *customisable packages* (using the stereotype «CustomisablePackage», cf. Fig. 6). Since packages allow to group an arbitrary number of arbitrary UML modelling elements together, the granularity of adaptation is not limited. Similar to classes providing customisation hooks for micro customisation within additional compartments, customisable packages provide customisation hooks within a pre-defined so called *customiser interface* (using the stereotype «CustomiserInterface», cf. Fig. 6).

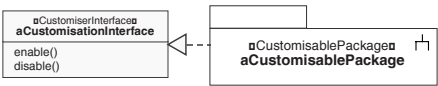


Fig. 6. Customiser Interface and Customisable Package

This customiser interface centrally provides customisation operations which are delegated to the appropriate classes of the customisable package. In that way, the customiser interface builds on customisation hooks defined during micro customisation. Besides pre-defined customisation operations such as `enable()` and `disable()`, the designer is able to specify own customisation operations, e.g., `switchToTextMode()`. Since for plain UML packages the specification of interfaces

is not allowed, we use in fact a special kind of package, namely *UML subsystems* (denoted by the fork symbol, cf. Fig. 6). In general, we consider such customisation packages as part of the variable part of an application representing different views on the stable part which are adapted by means of customisation rules.

5.2 Notation for Specifying Customisation Rules

Customisation rules can be seen as a uniform mechanism for both, macro adaptation and micro adaptation. For this, a customisation rule is simply specified within an annotation, using the stereotype «CustomisationRule» (cf. Fig. 7). Each customisation rule consists, according to the generic rule model, of a name and an ECA-triplet.

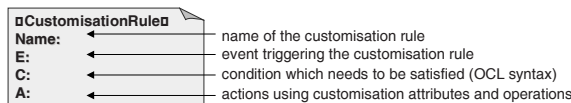


Fig. 7. Stereotyped Annotation for Specifying Customisation Rules

The annotation is attached to those model elements being subject to customisation, i.e., providing either the event for triggering the rule or the hook used by the action. According to our view of the customisation design process, customisation rules may be attached to model elements at content level, hyperbase level and presentation level [12]. The example depicted in Fig. 8 shows device-dependent customisation at the hyperbase level.

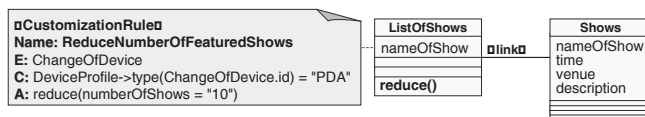


Fig. 8. Device-Dependent Customisation at the Hyperbase Level

Depending on the device used, the number of shows which are part of a single page "ListOfShows" is reduced to 10. For this, the event part of the rule incorporates the predefined event type *ChangeOfDevice*. Using profile information, the condition checks whether the device used was a PDA (note that additional rules would be needed to cover other device types).¹ The action uses the customisation operation *reduce()* to adapt the list of shows to the small device and consequently restricts the possible links to detailed information about shows.

5.3 Notation for Specifying Customisation Rule Patterns

To facilitate reusability of certain customisations, we introduce the notion of customisation rule patterns. Customisation rule patterns are some kind of

¹ The syntactically precise name of the user agent profile used in the condition would be *CustomisationModel::ProfileModel::UserAgent*. For sake of readability we have used the shortcut *UserAgent* instead.

parameterisable templates, predefining certain customisation rules in an application independent manner [13]. For their specification, the stereotype «CustomisationRulePattern» is used. Fig. 9 gives an example of a customisation rule pattern called *ReducerPattern* abstracting the device-dependent customisation rule of the example given in the previous section.

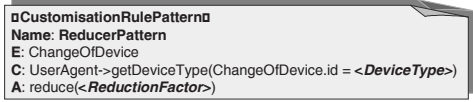


Fig. 9. ReducerPattern

The pattern reduces the number of list elements of a page according to a certain device type given. It takes two parameters namely the device type and the number of desired elements within the list.

6 Tool Support for Customisation

As any other modelling task within WUML, customisation design requires proper tool support. We propose an integrated modelling environment in terms of a *customisation toolkit* for the customisation design of ubiquitous web applications. Based on [20], this customisation toolkit should provide a *customisation rule designer* consisting of a set of graphical editors for defining and maintaining customisation rules, a *customisation rule browser* for facilitating the reuse of already existing customisation rules and an appropriate *repository* (cf. Fig. 10). Note that context modelling, profile modelling and the specification of customisation hooks should be supported by the tool constructed for designing the content level.

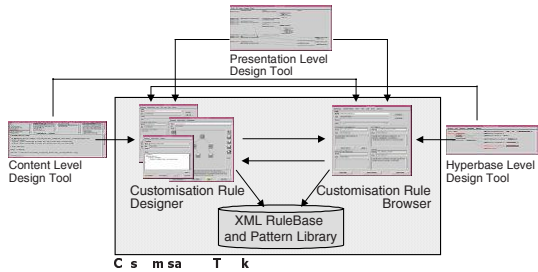


Fig. 10. Components of the Customisation Toolkit

It has to be emphasised that the components of the customisation toolkit should be *seamlessly integrated* among each other, as well as with the other tools provided by WUML. In particular, it should not only be possible to navigate between the tools as depicted by the arrows in Fig. 10, but also to preserve the context during navigation. This implies that, e.g., a composite event selected in the customisation rule browser is automatically loaded when an editor of the customisation rule designer is opened. Concerning the relationship to the other WUML tools, it is envisioned that the customisation toolkit can be opened directly within each of them in order to retrieve and optionally modify existing rules in the context of certain model elements or to

define new ones and attach them immediately to the appropriate model element using the annotations described in Section 5. In the following, the envisioned components of the customisation toolkit are discussed in more detail.

Customisation Rule Designer. The customisation rule designer should allow for a graphical specification and modification of rules, together with their components. The customisation rule designer provides different editors, according to the components of a customisation rule (cf. Section 4.3). Specified rules are automatically translated into an appropriate XML specification [26], and stored within the *rule base* of the repository (cf. Fig. 10). Besides rules, the customisation rule designer should also support the definition and modification of customisation rule patterns and the semi-automatic generation of rules out of them.

Customisation Rule Browser. The customisation rule browser should support the designer of the ubiquitous web application in retrieving existing rules or components thereof. One of the major design goals of the browser is to provide an interface that can be used rather intuitively. In this respect, the browser should be able to answer the following queries, to mention just a few:

- Retrieve all rules *realising a requirement*, i.e., having a certain ECA combination
- Retrieve all *components of a certain rule*
- Retrieve all rules, that are *triggered by a certain event* (e.g. change of location)
- Retrieve all rules, that contain a *certain condition/action*
- Retrieve all *actions*, that might be executed as soon as a *certain event is signalled*
- Retrieve all rules realising customisation within, e.g., *presentation design*
- Retrieve all rules using the *customisation hooks* of a *certain model element*

Customisation Rule/Pattern Repository. The customisation rule/pattern repository and the envisioned process of working with them is shown in Fig. 11.

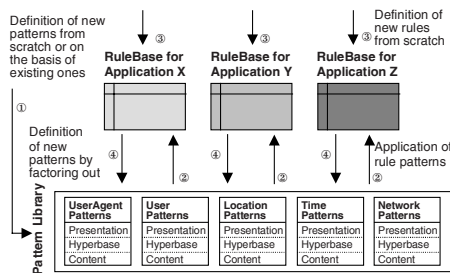


Fig. 11. Customisation Rule/Pattern Repository

In the long run, for each kind of customisation, appropriate rule patterns should be provided within a pattern library. The customisation rule bases and the customisation pattern library should be both organised as a set of dictionaries wherein all components are stored in terms of XML files. An important goal is to make the pattern library extensible allowing both, to define new patterns and to specialise existing ones (① in Fig. 11). At the same time, existing patterns or parts of them should be easily reused. In order to use rule patterns within an application, they have to be configured by the ubiquitous application designer. This is done by binding the parameters of a selected pattern on the basis of the application semantics. The system should guide the application designer during the process of parameter binding by providing on-line help for each required parameter and by restricting the binding

alternatives to those that do not contradict the specification of the underlying pattern. Once a rule has been fully and correctly specified, it can be automatically generated and stored in the rule base attached to the corresponding application (② in Fig. 11). Furthermore, it is still possible to specify rules without using a pattern and to store them directly within the appropriate rule base (③ in Fig. 11). This is normally done for rules without reusability in mind. If sometimes later an application designer recognises that a specific design situation recurs and thus is worth to be specified by a corresponding rule pattern, existing rules can be used for this abstraction process (④ in Fig. 11).

7 Outlook

In this paper we have introduced customisation as uniform mechanism for dealing with ubiquitous issues in web application development. Several issues remain to be resolved. We stress three of them. First, what is the "right" conceptual model of complex context and profile information? There are several standardisation efforts under way, such as CC/PP [24] and P3P [25], but to the best of our knowledge, there exists no comprehensive though open framework for context and profile modelling. In particular, we regard openness as crucial issue, since the kinds of relevant context and profile information will be growing in the near future (think of temperature in household systems, blood pressure in health systems, etc). Second, will there be an automatic recognition and computation of semantic equivalence? Which kind of modelling support is necessary? Resolving this issue is crucial for the success of "real" multi-delivery systems, where one can seamlessly switch from one delivery platform to another. Last but not least, where does customisation stop and redesign start? Since we are living in a changing world with changing requirements, continuous adaptation is almost a must. However, experience teaches us that continuously changing a software system will ultimately result in a complex inscrutable system. To find the right borderline between customisation and redesign is a challenging task.

Acknowledgements. This work was partially funded by *UWA (Ubiquitous Web Applications)*, an EU-funded Fifth Framework Programme project (IST-2000-25131) and *CustWeb (Customizeable Web Applications)* a strategic research project of the *Software Competence Center Hagenberg*.

References

- [1] G. D. Abowd, Software Engineering Issues for Ubiquitous Computing, International Conference on Software Engineering (ICSE), Los Angeles, 1999.
- [2] C. Avery, and R. Zeckhauser, Recommender Systems for Evaluating Computer Messages, Communications of the ACM (CACM), Vol. 40, No. 3, March 1997.
- [3] C. Barry, M. Lang, A Survey of Multimedia and Web Development Techniques and Methodology Usage, IEEE Multimedia, Special Issue on Web Engineering, April, 2001.
- [4] P. Brusilovsky, Adaptive Hypermedia: An Attempt to Analyse and Generalize, Multimedia, Hypermedia, and Virtual Reality: Models, Systems, and Applications, P. Brusilovsky, P. Kommers, and N. Streitz (eds.), Springer. Berlin, 1996.
- [5] J. M. Carroll, and A. P. Aaronson, Learning by Doing With Simulated Intelligent Help, Communications of the ACM (CACM), Vol. 31, No. 9, September 1988.

- [6] P. De Bra, Design Issues in Adaptive Web-Site Development, Proc. of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW, Toronto, Canada, 1999.
- [7] P. Fraternali, Tools and approaches for data-intensive Web applications: A survey, *ACM Computing Surveys*, Vol. 31, No. 3, September 1999.
- [8] J. Gomez, C. Cachero, O. Pastor, Conceptual Modeling of Device-Independent Web Applications, *IEEE Multimedia*, Special Issue on Web Engineering, April-June 2001.
- [9] M. D. Good, J. A. Whiteside, D. R. Wixon, S. J. Jones, Building a User-Derived Interface, *Communications of the ACM (CACM)*, Vol. 27, No. 10, October 1984.
- [10] G. Kappel, W. Retschitzegger, and B. Schröder, Enabling Technologies for Electronic Commerce, Proc. of the XV. IFIP World Computer Congress, Vienna/Austria and Budapest/Hungary, August/September 1998.
- [11] G. Kappel, W. Retschitzegger, The TriGS Active Object-Oriented Database System - An Overview, *ACM SIGMOD Record*, Vol. 27, No. 3, September 1998.
- [12] G. Kappel, W. Retschitzegger, W. Schwinger, Modeling Customizable Web Applications - A Requirement's Perspective, *International Conference on Digital Libraries: Research and Practice (ICDL)*, Koyoto, Japan, November 2000.
- [13] G. Kappel, S. Rausch-Schott, W. Retschitzegger, M. Sakkinen, Bottom-up design of active object-oriented databases, *Communications of the ACM (CACM)*, 44(4), 2001.
- [14] G. Kappel, W. Retschitzegger, W. Schwinger, A holistic view on web application development - the WUML approach, Tutorial notes at First Int. Workshop on Web-oriented Software Technology (IWWOST'01), Valencia, Spain, June 2001.
- [15] G. Kappel, B. Pröll, W. Retschitzegger, W. Schwinger T. Hofer, Modeling Ubiquitous Web Applications – A Comparison of Approaches, Proc. Of the Int. Conf. on Information Integration and Web-based Applications and Services (iiWAS), Austria, Sept. 2001.
- [16] B. Mobasher, R. Cooley, and J. Srivastava, Creating Adaptive Web Sites Through Usage-Based Clustering of URLs, Proc. of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX), November 1999.
- [17] R. Oppermann, and M. Specht, A Nomadic Information System for Adaptive Exhibition Guidance, Proc. of the International Conference on Hypermedia and Interactivity in Museums (ICHIM), D. Bearman and J. Trant (eds.), Washington, September 1999.
- [18] M. Perkowitz, and O. Etzioni: Towards Adaptive Web Sites: Conceptual Framework and Case Study, Proc. of the The Eighth Int. WWW Conference, Toronto, Canada, May 1999.
- [19] B. Pröll, W. Retschitzegger, R. R. Wagner, and A. Ebner, Beyond Traditional Tourism Information Systems - TIScover, *Journal of Information Technology and Tourism*, Vol. 1, Inaugural Volume, 1998.
- [20] W. Retschitzegger, TriGS Developer - A Development Environment for Active Object-Oriented Databases, 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI), Orlando/USA, July 2000.
- [21] W. Retschitzegger, and W. Schwinger, Towards Modelling of DataWeb Applications - A Requirements' Perspective, Proc. of the Americas Conference on Information Systems (AMCIS) Long Beach California, Vol. I, August 2000.
- [22] J. Rumbaugh, I. Jacobson, G. Booch, *The UML Ref. Manual*, Addison-Wesley, 1998.
- [23] M. Weiser, Some computer science issues in ubiquitous computing, *CACM*, 36 (7), 1993.
- [24] World Wide Web Consortium (W3C), Composite Capabilities/Preference. Profiles, <http://www.w3.org/Mobile>, 2001.
- [25] World Wide Web Consortium (W3C), Platform for Privacy Preferences (P3P) Project, <http://www.w3.org/P3P>, 2001.
- [26] World Wide Web Consortium (W3C), eXtensible Markup Language (XML), <http://www.w3.org/XML>, 2001.

Structuring Web Sites Using Audience Class Hierarchies

Sven Casteleyn and Olga De Troyer

Department of Computer Science, Vrije Universiteit Brussel,
Pleinlaan 2, 1050 Brussel, Belgium
svcastel@vub.ac.be, Olga.DeTroyer@vub.ac.be

Abstract. WSDM is an audience driven design method for web sites. By explicitly starting from the requirements of the web sites audience (the users or visitors), WSDM avoids problems caused by poor underlying design, or by a too data or organization driven view. This paper presents how the main structure of a web site can be derived from structuring the visitors of the web site into one or more so-called Audience Class Hierarchies. Each Hierarchy represents a classification of the visitors according to one aspect. The presented methodology forces the designer to deeply reflect on the requirements of the visitors, and to resolve any semantic conflict at design time. This greatly enhances the correctness of the obtained Audience Classes. The given algorithm also allows for computer-aided support of WSDM.

1 Introduction

As primary use of the Internet is more and more evolving to commercial purposes, an exploding amount of web sites and information is being offered through the World Wide Web (WWW) today. Most of these web sites are built without any underlying systematic design, instead, web designers rather focus on the graphical presentation of the site to be as hip or flashy as competitor's sites. Adding the intrinsic evolutionary nature of web sites, and their constantly changing information, current maintenance and usability problems become obvious (see [5] for a description of some problems). To address these problems, different design methods have been proposed: HDM and its successors HDM2 [9] and OOHDM [15] [16], RMM [11], W3DT [1] [2], SOHDM [13], WEBML [3] [4] and WSDM [5] [6] [7] [8] [10] [18]. WSDM uses an audience driven approach rather than a data driven approach: instead of letting the available data drive the design of the web site, WSDM uses the requirements of the intended users to drive the web site design process. This makes the Audience Modeling phase of WSDM a crucial phase in the design process.

The purpose of this paper is twofold. First we will describe the Audience Modeling phase of WSDM and introduce the concepts of "Audience Subclasses" and "aspect oriented Audience Class Hierarchies". Next, we present an algorithm to automatically derive the main structure of the web site starting from the different Audience Class Hierarchies. This algorithm also allows validating the Audience Class Hierarchies.

This paper is organized as follows: section 2 gives a short overview of WSDM. In section 3, Audience Classes and Subclasses are introduced. Section 4 introduces the advantage of different Audience Class Hierarchies for a single web site. Section 5 explains how the main structure of the web site can be derived from the Audience Class Hierarchy in case of a single hierarchy and in section 6 this is extended to the case of multiple Audience Class Hierarchies. Finally, section 7 gives conclusions.

2 WSDM: An Overview

The main characteristic of WSDM is the audience-driven approach. This means that instead of letting the structure of the available data set drive the design of the web site, as in most methods, we create a web site based on the requirements of the intended audience(s). In this way, WSDM gives consideration to the fact that web sites usually have different types of visitors that may have different needs.

A second important characteristic of WSDM is the distinction between the conceptual design (which is free from any implementation detail) and the design of the actual presentation: the grouping in pages, the use of menus, static and dynamic links, etc. This distinction is similar to the distinction made in database design between the conceptual schema (e.g. an E-R schema) and the logical schema (e.g. a relational schema). It allows making web site designs that are not biased by the diversity and rapid growing obsolescence of the web technology.

In figure 1 an overview of the WSDM method is given. The first step is to define the Mission Statement. The Mission Statement should express the purpose and the subject of the web site and declare the target audience. Based on this Mission Statement a two-step Audience Modeling phase is performed. In the first step, Audience Classification, the different kinds of users are identified and classified. Members of the same Audience Class have the same information and functional requirements. In the next step, Audience Class Characterization, the characteristics of the different Audience Classes are given. The result of the Audience Modeling is a set of Audience Classes together with an informal description of their information-, functional- as well as navigational- and the usability requirements, and their characteristics.

Next, we perform a Conceptual Design. The Conceptual Design phase is divided in three steps: Information Modeling, Functional Modeling and Navigational Design. During Information Modeling, Information Chunks are created. These chunks model the information requirements of the different Audience Classes. The different Information Chunks are linked together by a single information model, called the Business Information Model. All Information Chunks are defined as views on this model. In this way, possible redundancy is described and therefore can be controlled. During Functional Modeling, the functionality needed for the different Audience Classes is described. This is done using Functional Chunks. During Navigation Design we describe the (conceptual) structure of the web site and model how the members from the different Audience Classes will be able to navigate through the site. For each Audience Class a Navigation Track is created. Navigational

requirements are taken into consideration in this step. All Navigation Tracks together form the Navigation Model of the site.

The integration of the Information Chunks and Functional Chunks in the Navigation Model is called the Conceptual Model of the web site.

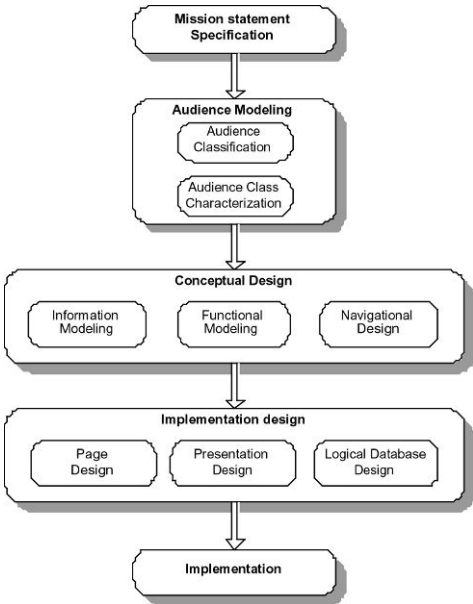


Fig. 1. WSDM overview

During Implementation Design, the (page) structure as well as the ‘look and feel’ of the web site is designed. The aim is to create a consistent, pleasing and efficient look and feel for the conceptual design by taking into consideration the usability requirements and characteristics of the Audience Classes. The design of the page structure (the grouping of information in pages) starts from the Navigation Model. If (some) information provided by the web site will be maintained by means of a database then the Implementation Design also includes the Logical Design of this database (derivable from the Business Information Model).

The last phase, Implementation, is the actual realization of the web site using the chosen implementation environment, e.g. HTML or XML. Depending on the complexity of the web site, parts of it can be automated using available tools and environments for assisting in HTML or XML implementations.

3 Audience Classes and Subclasses

We will now consider into more detail the second phase, the Audience Modeling Phase. To obtain the Audience Classes for a web site, WSDM looks at the activities of the organization that are related to the purpose and subject of the web site. Each activity involves people, which are potential users of the site if they belong to the

target audience of the mission statement. If necessary, the activities are decomposed in order to refine in each decomposition step the target audience. By definition, users belonging to the same audience class have the same (information and functional) requirements. Whenever the requirements differ, a new Audience Class is made.

Definition: an *Audience Class* is a group of potential visitors that belongs to the target audience of the mission statement, and has the same information and functional requirements. Graphically, we represent an Audience Class A as follows (figure 2):



Fig. 2. Graphical Representation of an Audience Class

We illustrate this process with an example: the Conference Review System (for the complete specification of this case study see [14]). The mission statement for this site could be formulated as follows: *“To support the overall selection process (submission by authors, evaluation and selection by the Program Committee) of papers for a conference”*. So, the purpose of the site is to support the paper selection process; the target audiences are authors and PC-program Committee; and the subject of the web site is papers for a conference. To refine this target audience into Audience Classes we look at the activities related to the purpose and subject of the web site, and examine the people who are involved in these activities. For the Conference Review System, the activities are Paper Submission, Assignment of Papers to PC Members, Assignment of Papers to Reviewers, Entering a review, Selecting Papers, Notify Authors (see figure 3). The people involved in these activities are Authors, PC Chair, PC Members, and Reviewers. To decide whether these can be one Audience Class or we need several Audience Classes, we look to their requirements. Due to lack of space, we have simplified the example to the most important requirements.

Authors

Functional requirements: submit paper, change submission, pre-register co-author
Information requirements: information about own submission

PC Chair

Functional requirements: create conference, pre-register PC Member, pre-register Reviewer, assign papers to PC Member, mark paper as accepted/rejected
Information requirements: view all available information

PC Members

Functional requirements: pre-register Reviewer, re-assign paper to reviewer, download papers assigned to him, submit review, advice PC Chair.
Information requirements: list of papers, view own reviews, state of reviews of other reviewers of papers assigned to him

Reviewer

Functional requirements: download papers assigned to him, submit review
Information requirements: view own reviews

The requirements for these different groups are sufficiently different to put them in separate Audience Classes. This results in four different Audience Classes: Authors,

PC Chairs, PC Members and Reviewers. However, notice that the set of requirements of Reviewer is a subset of the requirements of PC Member. Actually, a Reviewer needs to do exactly the same things as a PC Member, but in addition, PC Members have some extra needs: e.g. a PC Member has to be able to assign papers to Reviewers, indicate his preferences for some tracks/topics. Such a situation appears quite common in web modeling. Therefore, in analogy to the superclass-subclass relationship in OO, we have introduced the concept of “Audience Subclass”.

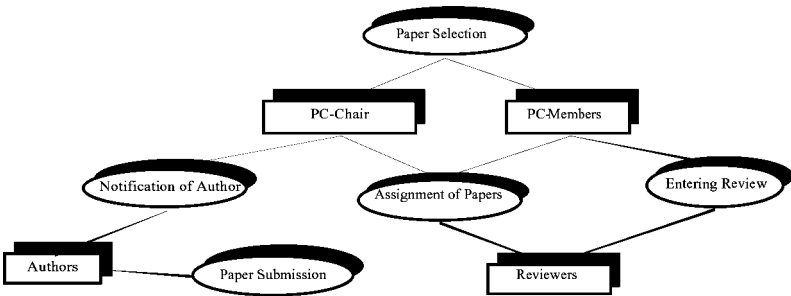


Fig. 3. Activity diagram for the Conference Review System

Definition: An Audience Class B is an *Audience Subclass* of an Audience Class A if B has all the requirements of A and some extra. Graphically, we represent the audience super/sub classes as follows (figure 4):



Fig. 4. Audience Subclass

In this way, we can create an Audience Class Hierarchy. Similar as in some OO programming languages, the Audience Hierarchy has a single top. In WSDM this common superclass is called *Visitor*. The Audience Class Visitor represents all potential users of the web site, including those that accidentally comes to the web site, and have no specific needs. Every Audience Class is a subclass of Visitor.

Considering the example of the Conference Review System, we have identified an audience subclass relationship between Reviewer (parent) and PC Member (child). The complete Audience Class Hierarchy (identified so far) is given in figure 5.

This method is based on the fact that we are able to identify (at a high level) the requirements of people involved in an activity. For projects where we exactly know the people involved in the activities and where we can involve (a selection of) this people in the development process (like for intranets), we can use the standard techniques of software engineering like questionnaires, interviews, etc. to collect the requirements. For most public web sites we are usually unable to involve the target audience itself in the development process. Therefore it looks as if we have to "guess"

for their requirements. Studying the characteristics of the audience may help to formulate their requirements. Usually, once the system is implemented and running feedback from the users will be needed to adjust and enhance the design.

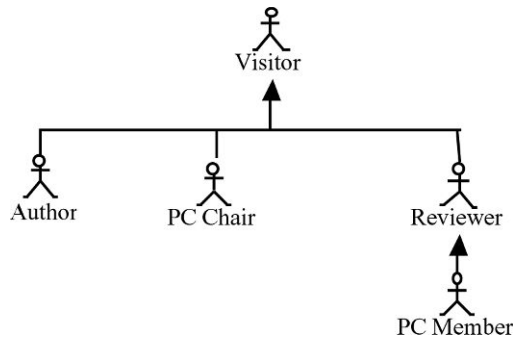


Fig. 5. Audience Class Hierarchy

Later on, during Navigational Design, a Navigation Track will be made for every Audience Class. Such a Navigation Track will fulfill all information- and functional requirements formulated for the Audience Class. Because for an Audience Subclass, part of its requirements is modeled at the level of its superclass, a "hierarchical" structure similar to the Audience Class Hierarchy will be present between the Navigation Tracks. This is explained into more detail in section 5.

4 Different Subclass Hierarchies

Until now, we neglected the fact that the Conference Review System needs to be protected against unauthorized use by means of a login. In addition, users can be pre-registered in which case they first need to confirm their registration. Using the same method as before, we can derive an(other) Audience Class Hierarchy only considering the requirements concerning these authorizations.

Registered Users

Authorization requirements: logging in

Pre-registered Users

Authorization requirements: logging in, confirm registration

Not-registered Users

Authorization requirements: register

This gives us the hierarchy of figure 6. We call this the Authorization Class Hierarchy because the focus in this hierarchy is the authorization requirements. Note that this hierarchy has also the class Visitor as its top.

Even though we could consider the authorization requirements together with the other requirements, we find it easier to separate them from the others and to concentrate on one aspect at a time. In this way several Audience Class Hierarchies can be

constructed, representing different aspects of the web site. Off course, for whatever web site, we always have at least one Audience Class Hierarchy. This kind of abstraction mechanism, used during modeling, can be compared in a sense to the concept of aspect oriented programming [12]. This is a very powerful concept and used in web design it allows simplifying modeling and results in easier specifications.

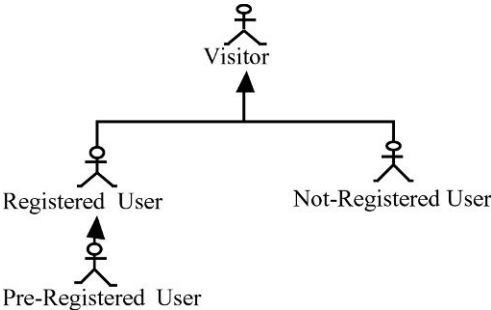


Fig. 6. Authorization Class Hierarchy

For some Audience Class Hierarchies, Audience Class Transitions can be specified. In the example of the Authorization Class Hierarchy, an Author may become a Registered User simply by registering; other users need to be Pre-Registered. The Audience Class Transitions for the example are given in figure 7.

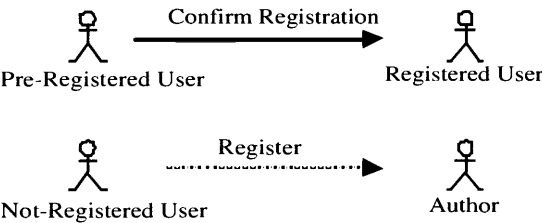


Fig. 7. Transition Diagrams for the Authorization Class Hierarchy

The introduction of different Audience Class Hierarchies has some impact on the construction of the Navigation Tracks during the Conceptual Design. But first, we describe the general principles for constructing the Navigation Schema.

5 Navigational Design

As already indicated in section 2, during Navigational Design we describe the (conceptual) structure of the web site and model how the members from the different Audience Classes will be able to navigate through the site. For each Audience Class a different *Navigation Track* is created. All Navigation Tracks together form the *Navigation Model*.

A Navigation Model is described in terms of *tracks*, *components* and *links*. Components represent units of information or functionality. They are connected by means of links. Links are used to model the structure of the web site as well as to indicate the need for navigation. We can put *conditions* on links to indicate that the availability of the link is dependent on the truth-value of the condition. Figure 8 gives the graphical notation for tracks, components and links. A multiple link is used to indicate that one component is linked to several instances of the other component.

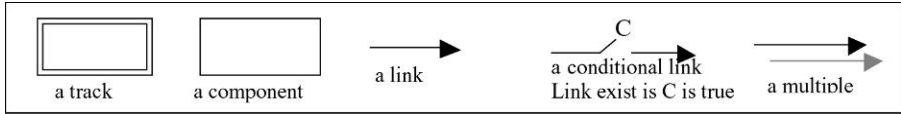


Fig. 8. Graphical Representation of Tracks, Components, and Links

The main structure of the Navigation Model can be derived from the different Audience Class Hierarchies. As already indicated, with each Audience Class will correspond a Navigation Track. To link the different tracks, the main idea is to follow the sublink structure of the different Audience Class Hierarchies. If there is only one audience class hierarchy, this is rather easy. The sublink structure of the Audience Class Hierarchy can be mapped in a one-to-one way into a track structure. Figure 9 shows the main track structure as derived from the general Audience Class Hierarchy of our Conference Review Site (given in figure 5). Note that for the sake of simplicity all links are represented as non-conditional links. The next step in the Navigational Design is to elaborate each Navigation Track into more detail. This is outside the scope of this paper; we refer to [7] for more details.

If there is more than one Audience Class Hierarchy, the translation into a Navigational Schema becomes more complicated. Actually, the different hierarchies need to be integrated (“weaved”) into a single hierarchy before we can map them into a Navigation Track structure. This will be explained in the next section.

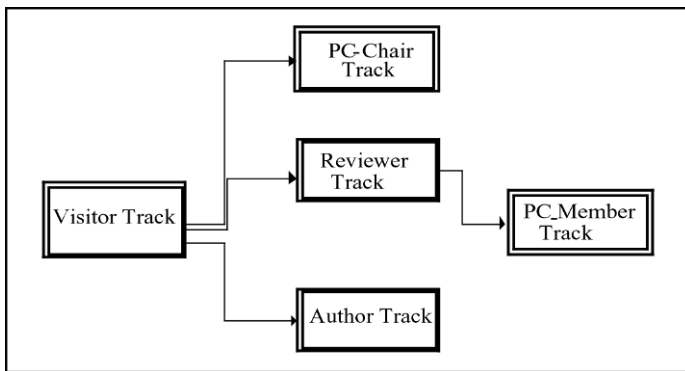


Fig. 9. Main Track Structure based on the General Audience Class Hierarchy

6 Mapping Audience Class Hierarchies into Navigation Tracks

The one-to-one mapping of an Audience Hierarchy into Navigation Tracks works fine as long as there is only one Audience Hierarchy. However, when other aspects come in, we need a way to merge ("weave") the different hierarchies into a single one before we can make the mapping. We will present an algorithm to do this¹. This algorithm is inspired by an algorithm given in [17], used to generate the object type structure for a conceptual schema. First we need some definitions.

Definition: An *Audience Class Matrix* M for a web site is an n by n matrix where each row i (with $0 < i \leq n$) of M is associated with some user requirement $UR[i]$, and each column i is associated with the same user requirement $UR[i]$. n is the total number of user requirements. The entries for the Audience Class Matrix M are 'Y' (yes) or 'N' (no), having the following meaning:

The entry on row i , column j is the answer to the question "Does every user who has the requirement $UR[i]$, also have the requirement $UR[j]$?"

Figure 10 presents the Audience Class Matrix for the example web site. For simplicity and lack of space, we have only considered the functional requirements and omitted the information requirements. As an example, the entry on row 1 and column 2 tells us that every user who can submit a paper can also change a submission. The entry on row 1 and column 4 tells us that not every user who can submit a paper can create a conference.

Evidently, the diagonal of any Audience Class Matrix will contain all 'Y' values. Also not all entries in the matrix are unrelated: if the question to UR 's (x,y) and (y,z) was yes, then also (x,z) must have an affirmative answer. In other words, the Audience Class Matrix is transitively closed. This is a useful proposition. After the designer has filled in the Audience Class Matrix we can check this property. If the matrix is not transitively closed it is usually due to some semantic errors.

Informally, every column i of an Audience Class Matrix represents which users can do $UR[i]$. E.g. column 1 represents all the users who are able to submit a paper. In this case the answer to the question "who may submit a paper?" is "evidently the users that can submit a paper, the users that can change a submission and the users that can pre-register a co-author". Column 13 represents the users who can login. If two or more columns are exactly alike, it means that the users that are represented by these columns are actually the same, and thus they belong to the same Audience Class. However, if the 'Y' entries of a column i is a subset of the 'Y' entries of another column j , this means that the set of users represented by column i will be a subset of the set of users represented by column j . Indeed, in the example, the users who may submit a paper (column 1) are a subset of the users who may login (column 13). Therefore we can define a subclass relationship between the Audience Classes represented by the columns i and j . In our example, the set of users that can submit a paper is a subclass of the set of users that can login.

¹ Actually, the algorithm does more than just merging Audience Classes. We will go into further detail at the end of the section.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Submit paper	Y	Y	Y	N	N	N	N	N	N	N	N	N	Y	N	N
2	Change submission	Y	Y	Y	N	N	N	N	N	N	N	N	N	Y	N	N
3	Pre-register co - author	Y	Y	Y	N	N	N	N	N	N	N	N	N	Y	N	N
4	Create conference	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
5	Pre-register PCMember	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
6	Pre-register Reviewer	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
7	Assign paper to PCMember	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
8	Mark Paper as accepted/rejected	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	Y	N	N
9	Re-assign paper to Reviewer	N	N	N	N	N	Y	N	N	Y	Y	Y	Y	Y	N	N
10	Download paper(s) assigned to review	N	N	N	N	N	N	N	N	N	Y	Y	N	Y	N	N
11	Submit review	N	N	N	N	N	N	N	N	N	Y	Y	N	Y	N	N
12	Advice PCChair	N	N	N	N	N	Y	N	N	Y	Y	Y	Y	Y	N	N
13	Login	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N
14	Confirm registration	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	N
15	Register	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y

Fig. 10. Audience Class Matrix

Formally, we define the above notions as follows:

Definition: For every Audience Class Matrix we can define a partial order relation ' $<$ ' on the columns of the matrix as follows:

column $i <$ column j if and only if for every row r , if column i contains a 'Y' on row r , then also column j contains a 'Y' on row r .

Informally, this means that for every row where column i has a Y, column j *must* also have a Y. In addition, column j may have Y's on rows where column i has N's.

Definition: Two columns i and j of an Audience Class Matrix are *equivalent*, if and only if column $i <$ column j and column $j <$ column i .

Equivalent columns are called *equivalent classes*, they form a potential Audience Class for the system.

Obviously, when column $i <$ column j , then the $<$ relation remains valid when we substitute column i by any of its equivalent columns, or column j by any of its equivalent columns. As a shortcut notation, we also use the $<$ relation between equivalent classes.

From the definition of Audience Subclasses, the Audience Class Matrix and the $<$ relation, we notice that the $<$ relation actually defines a subclass relation between the

potential Audience Classes. Column $i < \text{column } j$ implies that the Audience Class defined by the equivalence class of column i is a subclass of the Audience Class defined by the equivalence class of column j .

We can now give the algorithm to compute the integrated Audience Class Hierarchy:

Algorithm:

Step 1. Compute the Audience Class Matrix based on all the requirements formulated for the web site to be built.

Step 2. Compute the equivalence classes, and give them a meaningful name (these are the potential Audience Classes). The user requirements for an Audience Class are the requirements associated with the columns in the corresponding equivalence class

Step 3. Identify the ' $<$ ' relation between the different equivalence classes. *Step 4.* Compute the integrated Audience Class Hierarchy (using the Audience Classes and the ' $<$ ' relations between them).

We illustrate the algorithm for the (simplified) Conference Review System example.

Step 1: As already indicated, the Audience Class Matrix is given in figure 10. *Step 2:* The following equivalence classes are identified and given meaningful names. The names usually correspond with the names of the Audience Classes obtained during Audience Classification. However, new classes may appear (e.g. column 6):

- Columns 1, 2 and 3: Author
- Columns 4, 5, 7, 8: PC Chair
- Columns 9, 12: PC Member
- Columns 10, 11: Reviewer
- Column 13: Registered User
- Column 14: Pre-Registered User
- Column 15: Non Registered User
- Columns 6: User with Pre-register Reviewer right

Step 3: We identify the following $<$ relations:

PC Member $<$ Reviewer
 Author $<$ Registered User
 PC Chair $<$ Registered User
 Reviewer $<$ Registered User
 PC Member $<$ Registered User
 Pre-Registered User $<$ Registered User
 User with Pre-register Reviewer right $<$ Registered User
 PC Member $<$ User with Pre-register Reviewer right
 PC Chair $<$ User with Pre-register Reviewer right

Step 4: We notice that one of the potential Audience Classes, "User with Pre-register Reviewer right", was not yet identified before (in section 3 or 4). When we look to the requirements, indeed, we see that PC Chairs and PC Members actually have in common that they can both pre-register a Reviewer. However, as this is their only common requirement, this superclass is a rather artificial one. Therefore we decide not to take this class into consideration and to leave the pre-register requirement in both classes PC Chair and PC Member. The final integrated Audience Class Hierarchy is given in figure 11.

Having obtained the final integrated Audience Class Hierarchy, we can now, as described in section 5, use a simple one-to-one mapping to create the Navigation Track structure. For the example, this structure is given in figure 12.

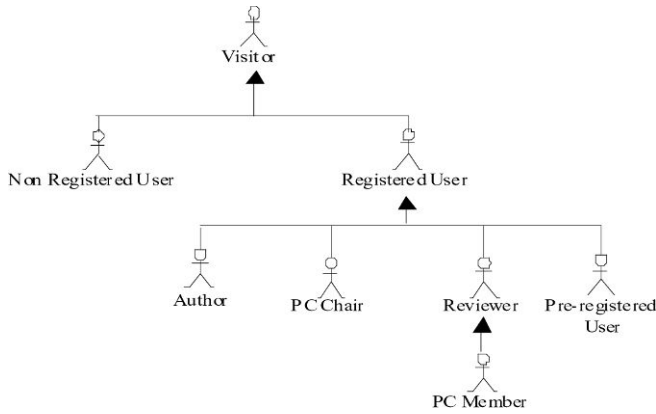


Fig. 11. Final Integrated Audience Class Hierarchy

However, so far we did not yet consider the Audience Class transition diagrams (given in section 4). These diagrams model dynamic transitions between Audience Classes. These can be mapped by defining a conditional link between the Navigation Tracks derived for the Audience classes involved. Figure 13 shows this mapping for the Transition Diagrams given in figure 7. Finally, figure 14 shows the complete Navigation Track Structure.

As can be seen from the example, the Audience Class Hierarchy Weaving algorithm does more than only merging different hierarchies. By using the requirements gathered for the different users as described in section 3, and by using these requirements to compute the Audience Class Matrix, the designer is forced to deeply reflect on these requirements. In the mean time, his work is validated, and can be corrected if necessary. In addition, the designer gets the integrated Audience Class Hierarchy, from which the Navigation Tracks can be derived easily.

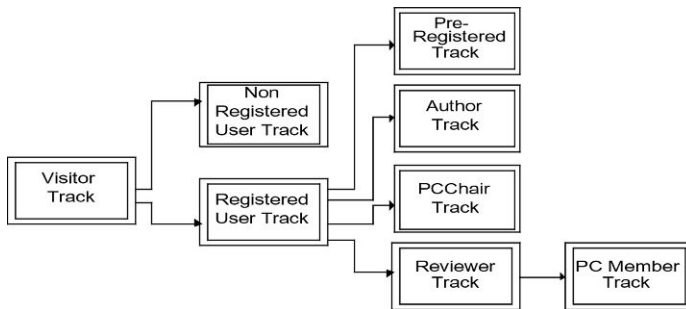


Fig. 12. Navigation Track Structure for the Conference Review System

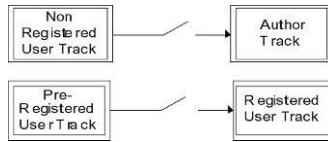


Fig. 13. Mapping Transition Diagrams into the Navigation Track

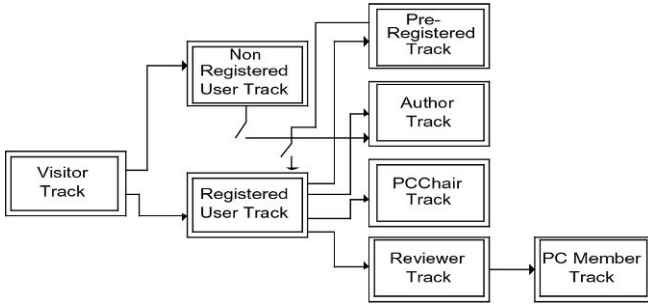


Fig. 14. Final Navigation Track Structure for the Conference Review System

7 Conclusions

In this paper we have explained and demonstrated a method that, based on the requirements of the different (potential) users (audiences), systematizes the design of the main structure of a web site. The method is defined in the scope of an audience driven web site design method, WSDM. The resulting structure of the web site is therefore based on the different Audience Classes identified for the web site. The Audience Classes can be identified in an aspect-oriented approach. This allows designers to concentrate on the user requirements of one aspect at a time. The different aspect oriented Audience Class Hierarchies derived in this way are weaved together using the Audience Class Weaving Algorithm. The method also allows some checks on consistencies and may detect ambiguities. In addition the designer is gently forced to precisely formulate the requirements of the different potential users of the web site. Implementation of the method seems feasible and would be very useful in the context of a Computer Aided Web Engineering (CAWE) tool for WSDM.

References

1. Bichler, M., Nusser, S.: "W3DT - The Structured Way of Developing WWW-Sites", Proceedings of the ECIS 1996, Lisbon, Portugal (1996).
2. Bieber, M., Isakowitz T.: "Designing Hypermedia Application", In Communications of the ACM, Volume 38, S. 26 - 29.
3. Bongio, A., Ceri, S., Fraternali, P., Maurino, A.: "Modeling Data Entry and Operations in WebML, Proceedings WebDB, p. 201-214, Dallas, USA (2000).

4. Ceri, S., Fraternali, P., A. Bongio: "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites", Proceedings of the WWW9 Conference, Amsterdam (2000).
5. De Troyer, O.M.F.: "Designing Well-Structured Web Site: Lessons to be Learned from Database Schema Methodology," In Proceedings of the ER'98 Conference, Lecture Notes in Computer Science (LNCS), Springer-Verlag (1998).
6. De Troyer, O.: "Audience-driven web design," In Information modelling in the new millennium, IDEA Group Publishing, ISBN 1-878289-77-2 (2001).
7. De Troyer, O., Casteleyn, S.: "The Conference Review System with WSDM", IWWOSt 2001, <http://www.dsic.upv.es/~west2001/iwwost01/>, Spain (2001).
8. De Troyer, O., Leune, C.: "WSDM: A User-Centered Design Method for Web Sites," In Computer Networks and ISDN systems, Proceedings of the 7th International World Wide Web Conference, Elsevier (1998), p. 85 – 94.
9. Garzotto, F., Paolini, P., Baresi, L.: "Supporting Reusable Web Design with HDM-Edit", In Proceedings of the 34th Hawaii International Conference on System Sciences, Maui USA (2001).
10. Goedefroy, W., Meersman, R., De Troyer, O.: "UR-WSDM: Adding User Requirement Granularity to Model Web Based Information Systems," In Proc. of 1st Workshop on Hypermedia Development, Pittsburgh USA (1998).
11. Isakowitz, T., Stohr, E.A., Balasubramanian: "RMM: A methodology for structured hypermedia design", Communications of the ACM 38, 8 (Aug 1995), p. 34-44.
12. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J., Irwin, J.: "Aspect Oriented Programming", Proc. Of ECOOP 1997, Springer- Verslag LNCS 1241, Finland (1997).
13. Lee, H., Lee, C., Yoo, C.: "A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems, Proc. Of HICSS (1998).
14. Schwabe, D.: "A Conference Review System" IWWOSt 2001, <http://www.dsic.upv.es/~west2001/iwwost01/>, Valencia, Spain (2001).
15. Schwabe, D., Guell, N., Vilain, P.: "Modeling Interactions and Navigation in Web Applications", Lecture Notes in Computer Science 1921, Proceedings of the World Wide Web and Conceptual Modeling 2000 Workshop, ER Conference, Springer, Salt Lake City (2000).
16. Schwabe, D., Rossi G.: "An Object Oriented Approach to Web-Based Application Design", Theory and Practice of Object Systems 4(4). Wiley and Sons, New York (1998).
17. Vermeir, D., Nijssen, G.M.: "A Procedure to Define the Object Type Structure of a Conceptual Schema", Inform. Systems Vol. 7 No. 4 p. 329-336 (1982).
18. WSDM website: <http://wsdm.vub.ac.be/>.

On the Automatic Extraction of Data from the Hidden Web

Stephen W. Liddle¹, Sai Ho Yau, and David W. Embley¹

Information Systems Group and Computer Science Department
Brigham Young University
Provo, UT 84602, USA
liddle@byu.edu, {yaus, embley}@cs.byu.edu

Abstract. An increasing amount of Web data is accessible only by filling out HTML forms to query an underlying data source. While this is most welcome from a user perspective (queries are easy and precise) and from a data management perspective (static pages need not be maintained; databases can be accessed directly), automated agents have greater difficulty accessing data behind forms. In this paper we present a method for automatically filling in forms to retrieve the associated dynamically generated pages. Using our approach automated agents can begin to systematically access portions of the “hidden Web.”

1 Introduction

The World Wide Web is not only vast, but it is growing at an enormous rate. Dealing with such huge quantities of information has been a challenge for both consumers and providers. To help manage this growth, it is becoming more common to use databases to generate Web pages dynamically.

Often, dynamically generated pages are accessible only through the use of an HTML form that invokes a Common Gateway Interaction (CGI) request to the Web server. CGI requests are often converted to database queries parameterized by information supplied by the end user through the supporting HTML form. Information available only through such CGI requests comprises a portion of what researchers sometimes call the “deep Web” [1] or the “hidden Web” [8, 20]. Unlike ordinary Web pages mapped to standard URL’s, information in the hidden Web is not accessible through regular HTTP GET requests by merely specifying a URL and receiving the desired page in response. Perhaps the information requires client authentication by means of a user ID and password. Or maybe the information is hidden behind a firewall in an intranet only accessible from particular IP addresses. Other portions of the Web are “hidden” only in the sense that none of the major search engines index those pages [13]. Most commonly, however, data in the hidden Web is stored in a database and is accessible by issuing queries guided by the aforementioned HTML forms.

A commercial vendor, BrightPlanet.com, claims the size of the deep Web is 500 times greater than the “shallow Web” [1]. Regardless of the actual relative

¹ Supported in part by the National Science Foundation under grant IIS-0083127.

sizes, it is clear that an enormous amount of data exists outside the so-called “indexable Web” [14]. Users want and need better access to this information.

Automated extraction of data behind form interfaces is desirable (1) when we wish to have automated agents search for particular items of information, (2) when we wish to wrap a site for higher level queries, and (3) when we wish to extract and integrate information from different sites.

We begin with a reference model of the information search task:

1. Formulate a query or task description.
2. Find sources that pertain to the task.
3. For each potentially useful source:
 - Fill in the source’s search form.
 - Analyze the results.
 - Gather any useful information that aids in the task.
 - Refine the query criteria and repeat if necessary.

Step 1 is often an informal step, but we will assume that the task is described clearly. Step 2, source discovery, typically begins either with a keyword search on one of the major search engines or a query to one of the directory services. Sometimes the user already knows where to begin or they have a social network that can quickly give a good recommendation [12]. For this study, we assume a potential source has already been discovered, and so limit our focus to step 3. In particular we seek to automate the filling in of forms, the analysis of results (including intelligent handling of errors), and the gathering of relevant data from the results. In the broader context of our system, information gathered during form processing will later be handed to a downstream data extraction process [7]; but data extraction is also beyond the scope of the current study.

1.1 Issues in Automatic Form Filling

There are many ways to design Web forms, and dealing with all the possibilities is not easy. Web forms can be built from a variety of controls such as radio buttons, checkboxes, selection lists, text boxes, hidden controls, and even author-defined objects. However, we can simplify this list because information transmitted to the server in a CGI request is fundamentally just a list of (name, value) pairs, appropriately encoded. Thus, we can characterize a form with n controls as a tuple $F = \langle U, (N_1, V_1), (N_2, V_2), \dots, (N_n, V_n) \rangle$, where U is the URL to which the encoded CGI request is sent, and the (N_i, V_i) are (name, value) pairs to be sent.

Unfortunately, this oversimplifies things with respect to the task of choosing the values to submit. There is considerable information associated with other metadata in the HTML form specification. For example, controls can be labeled as well as named; both labels and names might suggest possible domains we could associate with the fields. Text boxes often have a maximum content length (thus limiting the domain). A server might only respond to one of the two possible access methods (GET or POST), and finite domains for certain controls might be nicely specified (e.g., hidden controls have static scalar values, and radio

Type: Manufacturer:

Location: Year from: to

Price from to

Fig. 1. Typical Web Form from an Automobile Search Site

buttons, checkboxes, and selection groups each enumerate a relatively small set of possible values). There is a great deal we can learn from the constraints in the HTML, so it is very helpful when initially analyzing a Web form to consider the whole Document Object Model (DOM) representation of the form.

Some forms lead to other, more specialized forms that require further interaction. And it is often the case that a result page will include both retrieved data and a form for further processing. For example, when you search Google, there is a text box above the results that allows you to refine your search. Result pages might also contain error messages. Some error messages are easy to recognize automatically, such as an HTTP 404 error (page not found), which is defined as part of the HTTP protocol. Other error messages are more difficult to recognize automatically because the message is embedded within a series of tables, frames, or other types of HTML divisions. Users can usually understand these embedded messages quite easily, but automated understanding is more difficult.

Sometimes all the data behind a form can be retrieved with a single query. At other times, the data must be obtained piecemeal using multiple queries with different form control settings. When data is obtained piecemeal, we may retrieve duplicate information.

It might even be the case that interaction with the server requires a particular order of operation due to session tracking. For example, a login might be required, or cookies might be initialized and used to track a user's progress through a series of interactions with the server.

Furthermore, client-side scripts may interact with forms in arbitrary ways to modify and constrain form behavior. For instance, a text box control containing the total sales price in an order form might be automatically derived from the values in other text boxes by JavaScript whenever the form changes or is submitted to the server. JavaScript is often used to alter the behavior of forms. Unfortunately it is computationally hard to automatically analyze and understand arbitrary scripts.

All these issues present difficulties for automation. There are already parsers to convert HTML into a useful structured representation (i.e., DOM). But how can a system automatically fill in the fields of a form and submit it? How can a system deal with retrieved data, duplicate information, possible error messages or error notification pages, and embedded Web forms inside retrieved documents? Before describing our solution, we first discuss the related work.

1.2 Related Work

Others have also studied the problem of automatically filling out Web forms. Most common are tools designed to make it easier for an end user to fill out a form. For example, commercial services exist to provide information from a limited portfolio of user-specified information such as name, address, contact information, and credit card information [5,6,17]. These services, such as the Microsoft Passport and Wallet system, encrypt a user's personal information and then automatically fill in Web forms when fields can be recognized. Since many forms share common attributes (especially in the domain of e-commerce transactions), these tools can reliably assist users in entering personal information into Web forms.

One of the earliest efforts at automated form filling was the ShopBot project [4], which used domain-specific heuristics to fill out forms for the purpose of comparison shopping. Given the focus of this project, ShopBot did not propose a general-purpose mechanism for filling in forms for non-shopping domains.

More recently, researchers have considered the problem of assisting the user in a complex information search task that may span many Web sites and multiple Web forms. Davulcu et al. report on an architecture for designing "webbases" that help users perform complex domain-specific searches using a guided, by-example tool [3]. Underlying specifications are written declaratively by experts because webbases are probably too difficult for end users to create themselves. Some heuristics are mentioned, but few details are given regarding the actual process of filling out forms. This project appears to be a good attempt to simplify the creation of domain-specific search services, but it does not try to be a general-purpose crawler for the hidden Web.

There are now commercial ventures providing access to portions of the hidden Web. For example, BrightPlanet.com's Complete Planet as of December 21, 2000 claims to have indexed 38,500 databases containing deep Web content [2]. Another service, InvisibleWeb.com, claims to be "a directory of over 10,000 databases, archives, and search engines" [11] containing information from the hidden Web. Both of these commercial services claim to use semi-automated techniques for indexing the hidden Web, but they do not publicly discuss the details of their processes.

The most closely related work to our own is the Hidden Web Exposer (HiWE) project at Stanford [19,20]. Raghavan and Garcia-Molina propose a way to extend crawlers beyond the publicly indexable Web by giving them the capability to fill out Web forms automatically. Because of the formidable challenges to a fully automatic process, HiWE assumes crawls will be domain specific and human assisted (we also rely on human assistance at key points, but we do not currently use domain specific information in retrieving data from a particular site). Although HiWE must start with a user-provided description of the search task, HiWE learns from successfully extracted information and updates the task description database as it crawls. Besides an operational model of a hidden Web crawler, a significant contribution of this work is the label matching approach used to identify elements in a form based on layout position, not proximity within

the underlying HTML code. These researchers also present the details of several ranking heuristics together with metrics and experimental results that help evaluate the quality of the proposed process. The independently-developed details of our approach are complementary to HiWE. For example, we consider the task of duplicate record elimination and we use a stratified sampling approach to decide when a particular source has been fully extracted.

As a final note, we observe that many researchers have considered the problem of planning the information gathering process in a data integration context. HTML forms can be thought of as query templates with “binding patterns” [21] that limit the “access paths” to the data of interest [9]. Our purpose is slightly different, in that we are not given source descriptions ahead of time — we must automatically create suitable queries to retrieve the information behind forms. We also attempt to extract all of the available data at a particular source, not just answer a specific query at the source.

1.3 Outline

In the remainder of the paper we describe the details of our approach. We have created a prototype tool that allows users to semi-automatically and synergistically retrieve the data behind a particular source. In Section 2, we discuss how we fill out and submit a given form to a server with the object of retrieving all the data behind the form. Section 3 discusses our query execution plan, and the use of stratified sampling as a means to determine how much of the available data has likely been retrieved and how close we are to completion. Then in Section 4 we describe how we handle the issue of classifying the kind of result page returned by a query. Section 5 describes how we eliminate duplicate records from the accumulated results. We report on our implementation status, experimental results, and future work in Section 6

Recognizing that this project is a first step, we make several simplifying assumptions. First, we assume that user authentication is not required (or will be performed by the user who synergistically guides the search process), and forms of interest can be processed repeatedly, without regard to a user’s session state as maintained on the server. We also assume that client-side JavaScript embedded in HTML pages will not interfere with the form submission. In the event that a page contains multiple forms, we also assume that the first form is the one of interest. Finally, we assume that forms of interest will respond to submission via the HTTP GET method. Our initial experience suggests anecdotally that these assumptions are reasonable, but we will test this assertion quantitatively and more carefully in the coming months.

2 Automated Form Filling

The first step to automatically fill in and submit a chosen form is to parse the HTML page and extract useful information from the form description. So we begin by creating a parse tree for the given source page. An HTML form is

```

(1) <form action="/cgi-bin/umg/search.cgi" method="POST"
(2)     enctype="x-www-form-encoded">
(3) <b>Category:</b>
(4)   <select name="category" size="1">
(5)     <option selected value="">all categories</option>
(6)     <option value="accessories">accessories</option>
(7)     <option value="acoustic">acoustic (&quot;violin&quot;)</option>
(8)     <option value="drums">drums</option>
(9)     <option value="guitars">guitars</option>
(10)    <option value="keyboards">keyboards</option>
(11)    <option value="studio/stage">studio/stage</option>
(12)  </select> choose one<br>
(13) <b>Manufacturer:</b>
(14)   <input type="text" size="30" name="manuf"> example Gibson<br>
(15) <b>Model:</b>
(16)   <input type="text" size="30" name="model"><br>
(17) <b>Year:</b>
(18)   <input type="text" size="30" name="year"><br>
(19) <b>Condition:</b>
(20)   <input type="text" size="30" name="condition"><br>
(21) <b>Sort results in alphabetical order of:</b>
(22)   <select name="sort_by" size="1">
(23)     <option selected value="1">Category</option>
(24)     <option value="2">Manufacturer</option>
(25)     <option value="3">Model</option>
(26)     <option value="4">Condition</option>
(27)   </select><br>
(28)   <input type="reset" value="Clear Form">
(29)   <input type="submit" name="submit_search" value="SEARCH">
(30) </form>

```

Fig. 2. Excerpt of the Source Code of a Web Form (www.usedmusicgear.com)

indicated by the presence of start and end tags, `<form>` and `</form>` respectively [10]. If a form is present, we extract its portion of the parse tree and, for the purposes of experimentation and repetitive automated processing, we store the parse tree on disk in a easy-to-read form. Information of particular interest includes the source URL of the page, the action URL to which the form will be submitted, the number of fields, and details for each field, including field names, types (domain information such as the available options for a selection list), and default values. Figure 2 shows a cleaned up version of the HTML source for a Web form that retrieves information about used musical instruments. Note that for all the fields except *sort_by*, the default value is the empty string.

Given the parse tree for a form, we are ready to begin filling in field values. We start by assigning default values to each field. Once assignments are made, the form information needs to be encoded properly for the request method.

There are two ways to submit a form for CGI processing. First, using the HTTP POST verb, forms can be submitted with (name, value) pairs encoded in the body of the request. Second, using the HTTP GET verb, forms can be submitted by supplying the (name, value) pairs in the URL. A question mark (?) separates the base URL and action path from the encoded names and values. In the name/value list, an equality operator (=) separates a name from its assigned value, and an ampersand (&) separates one (name, value) pair from the next.

For the example form in Figure 2, the base URL is `http://www.usedmusicgear.com`, and the action path is `/cgi-bin/umg/search.cgi`. The form fields are *category*, *manuf*, *model*, *year*, *condition*, *sort_by*, *reset*, and *submit_search*. Using the default values from Figure 2, the value for *category* is blank (indicating

Category	Manuf	Model	Year	Condition	Price	Details
acoustic-woodwind	Emerson	P4	97	excellent	\$1100	Details
acoustic-woodwind	keilwerth	sxl1	1995	excellent	\$2200	Details
acoustic-brass trombone	Jupiter	JSL432S	1997	excellent	\$350	Details
acoustic-brass	King	Trombone	1996	excellent	\$550	Details
acoustic-euphonium	King	2280	1998	excellent	\$1600 /OBO	Details
acoustic-string cello	Czech, unknown	full size cello	circa 1900	excellent	\$2450	Details
acoustic-string cello	Karl Knilling	Cello	unknown	excellent	\$2110	Details
acoustic-string violin	Lark	Violin	1997	excellent	\$189	Details
acoustic-string violin	Tecchler, Daved	Master Grade	1707	excellent	\$9500	Details
acoustic-string	Vasil Valtchev	16 1/4" viola	1995	excellent	\$4500	Details
acoustic-string	Knilling	?		excellent	\$1200	Details
acoustic-woodwind alto sax	King	Professional	1994	good	\$750	Details
acoustic-woodwind alto sax	Yamaha	YAS-23	1996	excellent	\$720	Details
acoustic-woodwind alto sax	Yamaha			excellent	\$1050	Details
acoustic-woodwind alto sax			94	good	\$420	Details
acoustic-woodwind	Yamaha	62 Professional	1994	good	\$1750	Details
acoustic-woodwind	Selmer/Bundy	Baritone	??	excellent	\$1200	Details
acoustic-woodwind	Noblet	Professional	late 1950's	excellent	\$2200 OBO	Details
acoustic-woodwind	LeBlanc	Esprit	1996	excellent	\$800	Details
acoustic-woodwind	Yanagisawa	Professional	10 years	good	\$1200	Details

See the next 20 hits

Fig. 3. Returned Page Containing Retrieved Data

in this case “all categories”). So the (name, value) assignment for *category* is: *category*= . Similarly, the rest of the form field settings are assigned their default values, and the query is thus constructed as:

http://www.usedmusicgear.com/cgi-bin/umg/search.cgi?category=&manuf=&model=&year=&condition=&sort_by=1&submit_search=SEARCH

This query is then sent directly to the Web site of interest. It has the same effect as that of a user clicking the search button without selecting or typing anything on the Web form. Figure 3 shows the returned page for this query.

We can construct other queries by selecting various combinations of selection-list values, radio-button settings, and check-box selections. Usually it is not necessary to fill in text boxes automatically. Our system allows a user to provide values for text boxes, but does not require that values be provided. For forms with text boxes, our system only submits queries that have no entries for text boxes or that have user-supplied text-box values. The text boxes in the query that returned the results in Figure 3, for example, were all empty.

3 Query Submission Plan

Our goal is to retrieve all the data within the scope of a particular Web form. One way to do this is to fill in the form in all possible ways. Ignoring the issue of fields with unbounded domains (i.e., text boxes), there are still two problems with this strategy. First, the process may be time consuming. Second, we may have retrieved all the data before submitting all the queries. The latter case is often due to a default query that obtains all available data in the first place.

If one query or a small number of queries can retrieve all the data, it is desirable to quit before exhaustively processing all the combinations of the form fields in order to get all the information. We cannot be sure, however, that one query will always obtain all the information. Hence, we send a sampling of queries and determine statistically whether we have obtained all available data.

Table 1. A Two-Way Layout with Random Sampling

Factor B: sort by	Factor A: category							
		all categories	accessories	acoustic	drums	guitars	keyboards	studio/stage
	Category	x	x		x			
	Manufacturer		x					
	Model							
	Condition					x		

Table 2. Stratified Sampling

Factor B: sort by	Factor A: category (Regular Pattern)							
		all categories	accessories	acoustic	drums	guitars	keyboards	studio/stage
	Category	x				x		
	Manufacturer		x					
	Model			x				
	Condition				x			

Factor B: sort by	Factor A: category (Stratified Random Pattern)							
		all categories	accessories	acoustic	drums	guitars	keyboards	studio/stage
	Category	x					x	
	Manufacturer			x				
	Model							x
	Condition					x		

One way is to randomly select the sampling of queries among all possible queries. Unfortunately, randomly selected queries might not thoroughly and evenly cover all the fields in a form. Some fields may be underused while others are overused. The size of the sampling is another issue. How many samples should be good enough to determine whether we have retrieved all available data? In order to have better coverage and an adequate number of representative sample queries, we adopt a stratified sampling method [23,24].

3.1 Stratified Sampling Phase

The particular approach we take is to adopt the factorial and fractional techniques [23,16,18] of the stratified sampling method. The essence of the method is to avoid the probability of uneven selection of queries that might be biased toward certain fields. Stratified sampling seeks to use all form fields evenly to give maximum likely coverage of the underlying data.

The form in Figure 2 can be used to illustrate the formation of a stratified sampling pattern. Of the 8 controls in this form, we ignore the reset button and assume the submit button will always be selected. Further, 4 of the remaining controls are text boxes which we will ignore (if necessary, the user may choose values at runtime for the text boxes). Remaining are two selection lists, *sort.by* with 4 choices and *category* with 7 choices, that determine the set of queries we can attempt. Table 1 shows a two-way layout [23,15,16,18] that helps us use a two-factor method [23,16,18] to choose a query sample. In Table 1 we show a random sampling that is not stratified.

From Table 1, letting $p = 7$ and $g = 4$, we can observe that there are $p \times g = 28$ combinations to consider. In order to estimate how many sample queries should be used, we adapt the 2^k factorial experiments method [23,16]. Here k refers to the number of factors needed (*category* and *sort.by* in this example). We want


```

n                := number of sample queries from Equation 1
factor_A_list   := list of field names for factor A
length_A       := total number of field names for factor A
factor_B_list   := list of field names for factor B
length_B       := total number of field names for factor B
query_string   := a pre-constructed query with base URL, action path,
                  and criteria (like hidden fields) required for all queries.

While (n > 0) do:
  Randomly choose a field name from factor_A_list.
  Append the chosen field name as a criterion to query_string.
  length_A := length_A - 1
  If (length_A = 0) Then
    Reset factor_A_list to the list of all field names for factor A.
    Reset length_A to the number of fields for factor A.
  End if
  Randomly choose a field name from factor_B_list.
  Append the chosen field name as a criterion to query_string.
  length_B := length_B - 1
  If (length_B = 0) Then
    Reset factor_B_list to the list of all field names for factor B.
    Reset length_B to the number of fields for factor B.
  End if
  Send query_string to the target site and retrieve information.
  Reset query_string to its initial value.
  n := n - 1
End while

```

Fig. 4. Pseudocode for Stratified Sampling Method

$2^k = N$ where N is the total number of possible observations. Thus the desired sample size, n is defined by Equation 1:

$$n = \lceil \log_2 N \rceil \quad (1)$$

For the example in Figure 2, since $N = 28$ we have $n = \lceil \log_2 28 \rceil = \lceil 4.8 \rceil = 5$. Thus, our system first issues the default query (“all categories” from factor *A* and sort by category from factor *B*), and then an additional $n - 1 = 4$ queries.

Notice how the sample chosen in Table 1 does not cover three levels of factor *A* and one level of factor *B*. At the same time, “sort by category” and “accessories” are oversampled. Stratified sampling reduces this problem. Table 2 shows two stratified samples, in a regular pattern and a random pattern.

The regular pattern in Table 2 covers every level for factor *B* and most of the levels of Factor *A*. However, if the pattern is always chosen in this regular manner, a certain degree of bias is inevitably introduced since the left side of the table is always considered whereas the right side is not. On the other hand, the random pattern exhibits the advantage of thorough coverage and yet evenly distributes the selection randomly. Pseudocode of the algorithm for constructing the stratified sampling is shown in Figure 4. It randomly selects factors and levels, while also guaranteeing maximum coverage.

For three or more factors, the two-factor method can be generalized by replicating portions of factors over the columns and rows of the layout matrix [23]. The extension of the algorithm is straightforward so we do not present it here.

The number n obtained from Equation 1 is the upper bound of the stratified sampling size for determining the condition that if there is no new discovered information, the system should quit early and declare that all information has

been found instead of exhaustively submitting all N possible queries. During the submission of the n sample queries, the system keeps track of returned Web pages and whether the retrieved data has been seen before. In general, the data may be a subset of the first returned Web document resulting from the default query, it may be the same as the default, or it may be different. If no new information has been seen within n sample queries, it is assumed that the default query retrieved all information. If one of the returned Web pages has information that has never been seen before, then it is assumed that the default query has not retrieved all the information. In this case, the system enters to the exhaustive phase.

3.2 Exhaustive Phase

Before exhaustively executing all queries, we first estimate and report the maximum possible space needed for storing the results and the maximum remaining time needed to finish the process. The user can decide to continue the exhaustive query processing or stop and use only the information already obtained.

We estimate the maximum space requirement S by multiplying the total number of queries N by the average of the space needed for data retrieved from the n sample queries, as shown in Equation 2:

$$S = \left(\frac{N}{n}\right) \sum_{i=1}^n b_i \quad (2)$$

where b_i is the size in bytes of the i^{th} sample query.

We estimate the remaining time required T similarly, as shown in Equation 3:

$$T = \left(\frac{N}{n}\right) \sum_{i=1}^n t_i - \sum_{i=1}^n t_i = \left(\frac{N-n}{n}\right) \sum_{i=1}^n t_i \quad (3)$$

where t_i is the total duration of the i^{th} sample query. Note that we subtract the time already spent in the sample phase.

Finally we note that the user can specify a cutoff threshold to throttle query processing during the exhaustive phase if a certain number of consecutive queries have returned no new records. By default this threshold is the cardinality of the largest factor considered in the stratified sampling phase.

4 Processing a Query Response Page

Once a query is sent, the next step is to retrieve information from the target site. Seven different results are possible:

1. The response page may contain all the data behind the form.
2. The response page may contain data, but not show all the data for the query in a single page. Instead, there may be a “next” button or hyperlink leading to another page of data, such as the “See the next 20 hits” button in Figure 3.

3. The query might return data, but only part of the data behind the form because the query is just one of many possible combinations of the form fields.
4. The query may return a page that not only contains data, but also contains the original form.
5. The query may return a page that has another different form to fill in.
6. The query might return an error message or an error notification page, stating that certain text fields are required to be filled in, or simply a message stating that there is no record found for that submitted query.
7. Some other error cases might involve a server being down, an unexpected failure of a network connection, or other HTTP errors.

Our system maintains a timeout routine to terminate the operation if the case 7 or any abnormal delay occurs. In such a case the system reports the possible error(s) and aborts the current operation. When Web pages are being successfully retrieved within the timeout period, they are first saved as files in a temporary directory. The content of each returned page is then analyzed for the other 6 cases listed above. We now discuss each case in turn.

Response Page Contains All Data. This case occurs when the first default page already contains all the data specified within the scope of all possible combinations of form fields. The system determines that all the data might have been retrieved when consecutive returned pages from the sampling queries are either equal to or subsumed by the data on the first default page. Section 5 describes our approach to duplicate record detection. If all records in a response page are duplicates of data previously retrieved, we say the response page is subsumed. If all the sample queries return subsumed response pages, we terminate the query execution and decide that the first, default page did indeed contain all available data. Otherwise we treat the pages as instances of case 3.

Response Page Includes Next Link. Perhaps the most common case is that a Web site will return results a bit at a time, showing perhaps 10 or 20 results per page. Usually there is a link or a button to get to the next page until the last page is reached. For this case we treat all the consecutive next pages from the returned page as part of one single document by concatenating all the pages into one page. The system activates this process if the returned page contains a button or link indicating “next” or “more.” In this way, the system constructs a logical page containing all the data for the query. This logical page can then be treated as one of cases 1, 3, or 4.

Returned Page Contains Only Part of Site’s Data. This case is the common situation that every returned page may be some particular subset of the overall database. In this case, we can assume that even after all the sample queries have been processed, new information would still be available from the database behind the form. As explained previously, the system prompts the user asking whether to enter the exhaustive phase or discontinue the operation after the retrieval of the data from the sample queries. In this case, each response page is added to the repository we are building of retrieved data. Duplicate record elimination is described in Section 5.

Returned Data and Original Form Together. The system looks for a form structure within the response page. If found, the system compares the form in the returned page with the original form to see whether they are the same. If so, the system disregards the form in the returned page so that it will not recursively process the form indefinitely. Data in the response page is filtered for duplicates and added to the repository of results.

One Form Leads to Another. After all the sample queries have been processed, it may be the case that all the response pages contain another form to be filled in, but no data. We discover this case by comparing the structure of a response form with the original form. If they are equal or the original subsumes the response form, there is likely an error, so we treat it as case 6. Otherwise the system prompts the user for further action. One possibility is for the user to choose to process the new form, treating it as a new information source.

No-Record Notification or Required Field Missing. This situation occurs (A) when all the sample queries return no data or (B) required fields needed to be supplied by the user. In case (A), all the response pages would most likely contain the same message, e.g. “No matching records found”. In case (B), it is likely that all the response pages would contain the same “required field missing” message. Usually the original form is embedded in the error response page, perhaps with default values changed to reflect the query that failed. At this point our system prompts the user for intervention. The user either aborts the operation or supplies values for certain fields and the system repeats the sampling phase with the new values.

5 Filtering Duplicate Records

As we query a Web site, we gather retrieved records into a repository for later downstream data extraction. To avoid wasting space and to help us know when we have retrieved all the data behind a form, we eliminate duplicate records before placing them in the repository. To do this, we use the copy detection system [22], which detects sentence boundaries and computes hash values for each sentence. The copy detection system is highly effective for finding duplicate sentences over a very large set of textual documents. Instead of sentence boundaries, we wish to detect record boundaries, but it turns out that the two problems are very similar.

Records in response pages are usually displayed as paragraphs separated by the HTML paragraph tag `<p>`, rows in a table separated by `<tr>` `</tr>` tags, or blocks of data separated by the `<hr>` horizontal rule tag. In order to adapt the copy detection system for a collection of records, we devised a special tag called the sentence boundary separator tag denoted by `<s.>`. We then modified the copy detection system to acknowledge this special tag as the end of a sentence (i.e. the end of a record). During the duplicate filtering process, this tag is inserted into the retrieved Web documents around certain HTML tags which can most likely be discerned as the end of a record. The tags we have chosen for this treatment include `</tr>`, `<hr>`, `<p>`, `</table>`, `</blockquote>` and

`</html>`. If none of the above tags except `</html>` appears in the document, the whole document is considered to be a single record.

With this modification, the copy detection system is invoked. The detection system computes hash values for every record separated by `<s.>`. These hash values are compared with the hash values of all the records retrieved previously and stored in the repository. If duplicate records are found, they are eliminated. Unique records and their hash values are added to the repository.

6 Conclusion

In this paper we have described our domain-independent approach for automatically retrieving the data behind a given Web form. We have prototyped a synergistic tool that brings the user into the process when an automatic decision is hard to make. We use a two-phase approach to gathering data: first we sample the responses from the Web site of interest and then if necessary we methodically try all possible queries (until either we believe we have arrived at a fixpoint of retrieved data or we have exhausted all possible queries).

We have created a prototype (mostly in Java, but also using JavaScript, PHP, and Perl) to test our ideas, and the preliminary results are encouraging. We have been successful with a number of Web sites (e.g., www.autotrader.com, www.deseretnews.com, www.usedmusicgear.com), and we are now preparing a more comprehensive experiment. We will test 50-100 sites, gathering data on estimated and actual time to process, estimated and actual storage space required, total number of records retrieved, number of duplicate records reported, and manually verified actual number of duplicate records. We also plan to gather aggregate data indicating the number of sites visited, the number where our approach was successful, and how many required only the sampling phase. For those sampled, we will manually verify whether the result was complete. We will also measure the metrics suggested for HiWE, strict and lenient forms of submission efficiency. These are respectively ratios of successful (SE_{strict}) or valid ($SE_{lenient}$) queries to the total number of queries submitted [20].

Our research group is generally working in the broader context of ontology-based data extraction and information integration. If we combine the hidden Web retrieval problem with the tool of domain-specific ontologies, we could automatically fill in text boxes with values from the ontologies. While this makes the retrieval process task-specific, it also increases the likelihood of being able to extract just the relevant subset of data at a particular Web site. (Our assumption in this paper has been that the user wants to retrieve all the data at a given site.) Also, it is likely that we could avoid a fair amount of the manual intervention in our current process if we use ontologies.

Users need and want better access to the hidden Web. We believe this will be an increasingly important and fertile area to explore. This paper represents a step in that direction, but a great deal remains to be done. We look forward to continuing this promising line of research.

References

1. Michael K. Bergman. *The Deep Web: Surfacing Hidden Value*. BrightPlanet.com, July 2000. Downloadable from http://www.brightplanet.com/deep_content/-deepwebwhitepaper.pdf, checked August 10, 2001.
2. Completeplanet.com home page. <http://www.completeplanet.com>. Checked August 10, 2001.
3. Hasan Davulcu, Juliana Freire, Michael Kifer, and I.V. Ramakrishnan. A layered architecture for querying dynamic Web content. In *SIGMOD '99 Proceedings*, pages 491–502, Philadelphia, PA, May 1999.
4. Robert B. Doorenbos, Oren Etzioni, and Daniel S. Weld. A scalable comparison-shopping agent for the World-Wide Web. In *Proceedings of the First International Conference on Autonomous Agents*, pages 39–48, Marina del Rey, CA, February 1997.
5. Patil systems home page. <http://www.patils.com>. Describes LiveFORM and ebCARD services. Checked August 10, 2001.
6. eCode.com home page. <http://www.eCode.com>. Checked August 10, 2001.
7. D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data and Knowledge Engineering*, 31:227–251, 1999.
8. Daniela Florescu, Alon Y. Levy, and Alberto O. Mendelzon. Database techniques for the World-Wide Web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.
9. Alon Y. Halevy. Answering queries using views: A survey. *VLDB Journal (online, to appear)*, 2001.
10. HTML 4.01 specification. <http://www.w3.org/TR/html4>, December 1999. Checked August 10, 2001.
11. InvisibleWeb.com home page. <http://www.invisibleweb.com>. Checked August 10, 2001.
12. Henry Kautz, Bart Selman, and Mehul Shah. The hidden web. *AI Magazine*, 18(2):27–36, Summer 1997.
13. Steve Lawrence and C. Lee Giles. Accessibility of information on the Web. *Nature*, 400:107–109, 1999.
14. Steve Lawrence and C. Lee Giles. Searching the World Wide Web. *Science*, 280:98–100, April 1999.
15. Thomas Leonard. *A Course In Categorical Data Analysis*. Chapman & Hall/CRC, New York, 2000.
16. Robert A. McLean and Virgil L. Anderson. *Applied Factorial and Fractional Designs*. Marcel Dekker, Inc., New York, 1984.
17. Microsoft Passport and Wallet services. <http://memberservices.passport.com>. Checked August 10, 2001.
18. R.L. Plackett. *The Analysis of Categorical Data, 2nd Edition*. Charles Griffin & Company Ltd., London, 1981.
19. Sriram Raghavan and Hector Garcia-Molina. Crawling the hidden Web. Technical Report 2000-36, Computer Science Department, Stanford University, December 2000. Available at <http://dbpubs.stanford.edu/pub/2000-36>.
20. Sriram Raghavan and Hector Garcia-Molina. Crawling the hidden Web. In *VLDB 2001 Proceedings*, Rome, Italy, September 2001. To appear.
21. Anand Rajaraman, Yehoshua Sagiv, and Jeffrey D. Ullman. Answering queries using templates with binding patterns. In *PODS '95 Proceedings*, pages 105–112, San Jose, CA, 1995.

22. Randy D. Smith. Copy detection system for digital documents. Master's thesis, Computer Science Department, Brigham Young University, 2000.
23. Ajit C. Tamhane and Dorothy D. Dunlop. *Statistics and Data Analysis: From Elementary to Intermediate*. Prentice-Hall, New Jersey, 2000.
24. Peter Tryfos. *Sampling Methods For Applied Research: Text and Cases*. Wiley, New York, 1996.

MIDAS/BD: A Methodological Framework for Web Database Design

E. Marcos, P. Cáceres, B. Vela, and J.M. Cavero

Kybele Research Group
Rey Juan Carlos University
Madrid (Spain)

{cuca, p.caceres, b.vela, jmcavero}@escet.urjc.es

Abstract. One of the most important trends in Information System research is concerning Web application development. Generally, Web Information Systems are developed *ad-hoc*, that is, without modeling them before their implementation. At best, developers use techniques and methodologies directly imported from the traditional Software Engineering field. However, classic methodologies do not fit to some key aspects of Web applications as, for example, navigation design. This is the reason why different modeling techniques and methodologies for Web Information Systems development have appeared. MIDAS is a methodological framework for Web Information System development that can be customized according to the type of application we want to develop (hypermedia applications, database applications, etc.). In this paper we present MIDAS/BD, the specific part of MIDAS for Web Database development. MIDAS/BD is based on XML and (object -) relational technology.

1 Introduction

Because of the popularity that the World Wide Web has acquired in the last years, nowadays, there are a large amount of software systems developed in connection with this technology. Web sites have become central repositories of information with a uniform interface to share data that is used by organizations in both, Internet and Intranet, environments. Until now, most of the Web applications are being developed, at best, applying traditional Software Engineering methods [15]. But, unfortunately, such methods do not take into account some important aspects such as the definition of the system architecture (physical and software architecture), hypertext design, data maintenance, security and so forth. Due to this fact, different methodologies for Web Information Systems (WIS) development have appeared in the last years [1,5,7,10,12,14,16,19,21].

However, each of the above mentioned proposals focus on a specific type of Web applications. So, [14,16,21] are some examples of hypermedia and multimedia design methodologies, and [1,5,10,19] are examples of methodologies for web site design and its maintenance.

With regard to the traditional methodologies, recently some changes are being proposed concerning the process model as well as new techniques for Web

applications modeling and design. Perhaps, the most known work in this sense is the UML extension for Web application modeling proposed by Conallen [6, 7]. There are other extensions of traditional methodologies such as OO-*H*method [12] or the Fournier proposal, a methodology for C/S and Web applications development [8]. Other recent approaches, tend to define generic methodologies for WIS development, integrating the previous mentioned perspectives and providing techniques for hypertext design, data design, workflow design, data warehouse design, etc. [3,5]. MIDAS is a generic methodology for WIS development. It proposes a process model [4] that can be customized by the designer according to the type of WIS that will be carried out. MIDAS distinguish between different kind of WISs according to a proposed classification .

The most interesting characteristic of MIDAS is that it is a *light* methodology which has been conceived according to the customers and developers necessities. Usually, introducing a new methodology in a company, implies that developers have to change their habitual way of working. Moreover, when customers order a software product, it is crucial to satisfy them. Generally they want to publish their business information in the Web as soon as possible. Due to the mentioned reasons, MIDAS has been conceived to fulfil the following objectives: to provide developers with a methodology that guides them in their work based on their usual way of working; to support a quick software development assuring the customers that a first version of the software will be available as soon as possible; to limit the quantity of documentation generated during the project.

This paper deals with MIDAS/BD, that focuses on the techniques and the method proposed for Web database modeling and design. The last phase of MIDAS methodology (MIDAS/FT), is not covered in this paper. MIDAS/BD is suitable to design WISs that are just databases in the Web and do not need to design complex functionality, workflow, etc. MIDAS/BD is based on (object-) relational and XML technologies.

The rest of the paper is organized as follows: section 2 is an overview of the MIDAS framework; section 3 is the proposal for Web database developing; in section 4 we sum up a case of study of the proposed methodology; finally, section 4 sums up the main conclusions as well as the future work.

2 Overview of MIDAS

MIDAS is a methodological framework that can be customized for different kind of WISs. It is based on the classification of WISs proposed by Cáceres and Marcos [4]. This is a simple classification that allows us to identify the activities that should be carried out. So, MIDAS distinguishes between *Web hypertext* applications (Web sites that provide just static and unstructured information), *Web database* applications (data intensive Web sites, with dynamic and structured information) and *Web complex functionality* applications (Web sites that offer services to the user as, for example, e-commerce applications). Table 1 shows the criteria that have to be into account in the development of each of these WISs.

Table 1. Criteria to consider for WIS development

WIS classification	WIS characteristics			
	External criteria		Internal criteria	
	Presentation	Navigation	Information structure	Logic of the application
Hypertext	+	+	-	-
Web databases	+	+	+	-
Complex functionality	+	+	+	+

A representative client of Web applications usually needs, as a primary requirement, that the software product was available as soon as possible. His main objective is to make Web users known his business information and services that will be provided through the Web (esta frase no la entiendo bien). If Web software developers want to be successful, they have to understand that the previous mentioned requirement is essential for the user. The process model of MIDAS includes the process characteristics for hypermedia applications [11]. So, MIDAS proposes an iterative and incremental process with prototyping. Besides, an important goal of MIDAS is to provide an agile process [2,9], which is essential for developing Web applications, as we explained above. So, MIDAS will provide user products in short times allowing to introduce, in each iteration, new requirements that have not been identified in the first steps of the project. Moreover, testing will be done in each iteration reducing in this way the risk of failure.

The MIDAS process model proposes to build the product in three increments according to the characterization of WISs previously defined (see Table 1). Although it is a simple classification it is enough to define the life cycle because other criteria, as security or architecture design, are orthogonal and have to be considered in every kind of WIS, if it is required. Considering these increments, and adding the requirement definition phase, MIDAS proposes developing the product in four phases as it is shown in figure 1.

The client will get a first version of the product at the end of the second phase. At this moment, the product will provide the user interface that allows navigating through static Web pages and no functionality will be implemented yet. As a result of the user utilization of this first version, at the end of the second phase the product will provide the refined user interface. The database will be also implemented and Web pages will be generated dynamically, extracting the information from the database. Complex functionality WISs will be finished after carrying out the four phases proposed by MIDAS. Working in this way, each version of the product provides a qualitative increment according to the WIS type.

Note, that we propose this sequence for the product development because we focus on Web products where it is supposed that a main requirement is to be in the Web as soon as possible. In other cases this process could not be appropriate. So, for example, in the development of a database that we will perhaps want to publish in the Web, the database should be developed before building the hypertext.

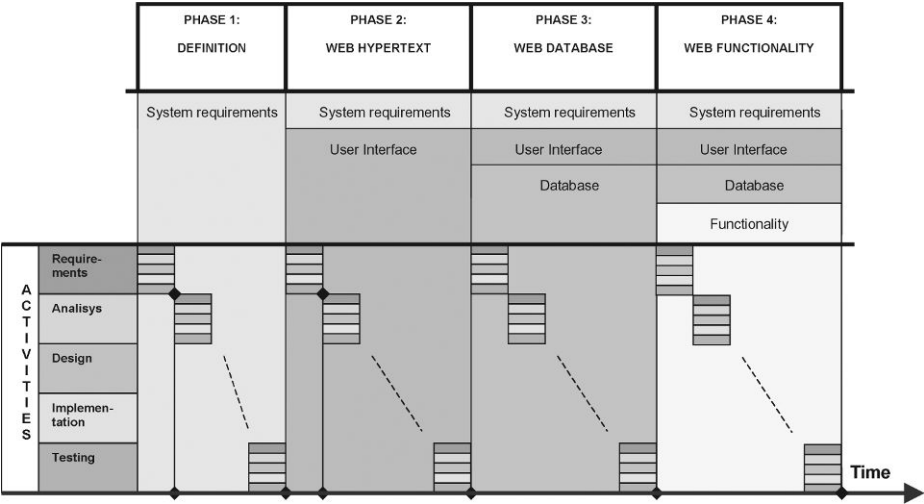


Fig. 1. Process model of MIDAS

3 Web Database Design

Web sites contain large quantities of data that are valuable sources of information for their owners and, however, they are not always rightly exploited. Usually, Web sites are organized in a casual and chaotic manner and this kind of organization gives rise to many problems in data maintenance task as, for example, problems finding and relating data, some mistakes or data inconsistencies, obsolete data, etc. To solve these problems, some methodologies for web site design and maintenance have been proposed [1,5,10,19].

According to the MIDAS process model we are going to explain in this section the process and techniques proposed in MIDAS to develop Web database applications. Table 2 shows for each phase and activity of MIDAS (see figure 1), the tasks and related techniques of MIDAS/BD as well as some proposed notations. The sequence to carry out the proposed tasks is shown in figure 2. Note that the MIDAS/BD process includes the MIDAS/HT phase, that is presentation and navigation design, since the Web database design have to take into account its Web user interface. We focus on the analysis, design and implementation activities due to the fact that requirements and testing do not need specific techniques for this phase that were different that the techniques used for these activities in the definition or complex functionality phases.

3.1 Web Hypertext Phase

The Web Hypertext phase is the second phase in the development of a Web database and it is carried out after studying the user requirements. This phase has two main *objectives*: a) to understand better the specific requirement of the user by building a prototype of the system; b) to provide the user a first version of the product in a short time.

Table 2. Overview of MIDAS for *Web data intensive* application development

Activity	2 th . Phase: Web HyperText (MIDAS/HT)			3 th . Phase: Web DataBase (MIDAS/BD)		
	Task	Technique	Notation	Task	Technique	Notation
ANALYSIS	Data Conceptual Design	Data Conceptual Model	Class D. (UML)	Refined Data Conceptual Design	Data Conceptual Model	Class D. (UML)
	Hypertext Conceptual Design	Slice Conceptual Model Navigation Conceptual Model	Slice Diagram & Navigation D. (RMM)	Refined Hypertext Conceptual Design	Slice Conceptual Model Navigation Conceptual Model	Slice Diagram & Navigation D. (RMM)
	Presentation Conceptual Design	Abstract Interface Model	AVD (OOHDM)	Refined Presentation Conceptual Design	Abstract Interface Model	AVD (OOHDM)
DESIGN	Hypertext Logical Design	Prototyping with Graphical Design Tools (Dream Weaver, Front Page, etc.)		Logical Data Design	(Object)-Relational Model	UML extensions
	Presentation Logical Design			Refined Hypertext Logical Design	Slice Logical Model Navigation Logical Model	DTD XML
				Refined Presentation Logical Design	Presentation Logical Model	XSL
IMPLEMENT.	User Interface Implementation		HTML	DB Implementation		Specific product SQL
				XML-SQL Integration		XML/ASP
				Refined User Interface Implementation		HTML/WML
						...

Analysis activity. In the analysis activity we have to obtain the conceptual models of the WIS. The first one to be realized has to be the data conceptual model that allows us to identify the persistent classes of the system. Afterwards, we will build the navigation conceptual model that represents how the data will be structured in the hypertext. In parallel to the navigation model we will define the abstract interface model that allows us to define how data will be presented (by a list or a button, if a picture should be shown etc.). Two last models together represent the user interface conceptual model. As an intermediate task between the data and navigation conceptual models, we propose to represent the slice conceptual model that represents the information grouped in the same way as it will be shown in the screen. It is similar to the slice diagram provides by RMM [13] but with some differences. This model will allow us representing the information extracted from the data conceptual model but represented grouped by screens. So, some classes will be sliced and the slices from different classes could be joined according to the presentation. The navigation conceptual model will represent how to navigate through the slices defined in the slice

conceptual model (identifying one-way or two-way links, including access index and so forth). As recommended notation to represent the different models of this activity, we propose to use the UML class diagram for the data conceptual design, because it is the standard notation for object-oriented developments. However, as UML does not provide any specific notation for hypertext modeling we propose to use techniques of specific methodologies for hypertext design as [13] or [20]. In our opinion, these techniques continue being better for Web hypertext design than the ones provided by the Conallen extension [7]. The slice diagram is adapted from the RMM slice diagram allowing to represent in each slice information coming from different classes. Another difference is that in our approach the slice diagram is unique, that is to say, there will not be a slice diagram for each class as in RMM. The presentation conceptual model could be done using the OOHDM AVD diagram [20].

Design activity. The design activity is an intermediate step between the analysis and the final implementation of this product version. In this step we have to define some details about the conceptual models of the Web database user interface: navigation and presentation logical models. In this phase we will not obtain the design of the database because this task will have to be carried out in the next phase. However, we have defined the conceptual model of the database because it provides clues to get the navigation conceptual model. So, in this activity we have to define the colors, fonts, icons, etc. We have also to redefine how links are implemented (one or two-way links), etc. The proposed technique in this phase is to develop different prototypes using some graphical design tools (as for example Dream Weaver, Front Page, etc.).

Implementation activity. Due to these kind of tools generate HTML code, the last prototype is the final version of this phase. So, the Web user interface implementation of the database will be done in HTML generated with a graphical design tool.

3.2 Web Database Phase

The Web Database phase is the third phase in the development of a Web database and it is carried out after developing the first version of hypertext. This hypertext will be the Web user interface of the database. However, at this moment the pages are not dynamically generated that is, no information is retrieved of the database. The *objective* of this phase is to build the database and to implement a new version of its Web user interface that was dynamically generated extracting the data of the database. New user requirements, coming from specific database requirements, could be also considered. In this phase any functionality of the application, except the required functionality for the database (insert, delete, update and select operations) will be considered. Figure 2 show the process to get the final Web database.

Analysis activity. In the analysis activity the conceptual models that have been obtained in the previous phase will be refined taking into account new user requirements. These requirements are obtained on the one hand from new specifications given for the database definition and on the other hand from the feedback provided by the use of the prototype obtained at the end of the first phase.

Techniques and notations are the same of the ones proposed for the Web Hypertext phase.

Design activity. In the design activity of this phase we have to: a) obtain the logical design of the database and b) refine the logical design of the hypertext as well as of the presentation. We propose to use UML extensions [17] as the notation for the database logical design. The final Web user interface will be implemented using XML technology. So, the hypertext design in this phase will include both, the DTD and the XML documents, and the presentation design will be done defining the corresponding XSL documents.

Implementation activity. The final implementation includes the implementation of the database in the selected product and to integrate the XML and the database through ASP. Applying the corresponding XSL template we will get the HTML (or WML, etc.) pages with the dynamic information retrieved from the database.

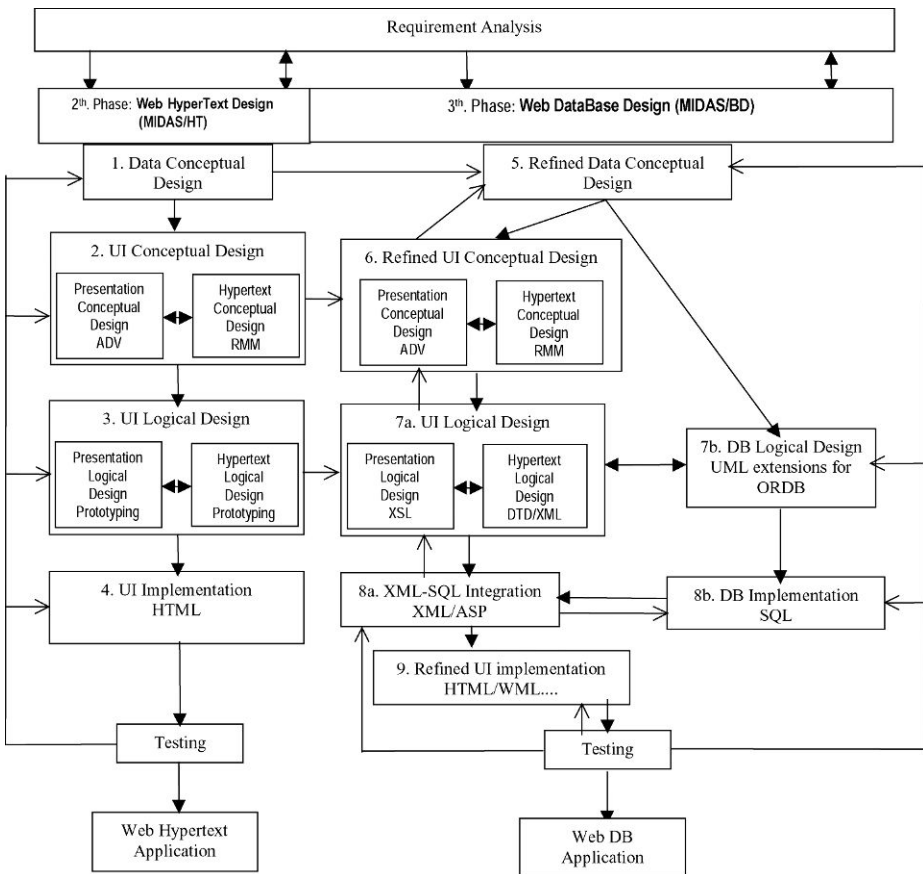


Fig. 2. Web Database development process proposed in MIDAS/BD

Although we propose notations for each technique, we think that it would be better to have a unique notation that allows modeling the whole system. However, this notation does not exist. There are some UML extensions for Web modeling [7] but they do not support all techniques that we have proposed to use. So, for example, this UML extension provides a basic navigation model and almost every navigation model of specific methodologies for hypertext design (as RMM or OOHD) are much more expressive. We have already defined an UML extension for object-relational database design [18] and we propose to use it in the step 7b (DB logical design). To pass from one step to another in the process it is convenient to provide some guidelines to the developer. In MIDAS/BD we have already defined the guidelines needed to pass from the fifth to the seventh step and from the seventh to the eighth step [17].

4 A Case of Study

We are validating the proposed methodological framework applying it to different WISs. To validate MIDAS/BD we are developing different Web database applications and in this section we present an example. It is a Web site with information about the *Kybele Research Group* where the information of the pages will be retrieved dynamically from an object-relational database.

In the first phase, we have developed a first version of the pages (you can visit the pages at kybele.escet.urjc.es) following the steps indicated in table 2 and figure 2 for MIDAS/HT. This version has been designed using Front Page which generates HTML code. These pages have static information and are not complete. So, for example, the English version or some final pages are not implemented. However, this prototype is enough to have the main information available in the Web. We have refined the conceptual models (the data, slice, navigation and presentation conceptual models) by designing these pages because we have detected new links, the colors that we prefer, as well as some information that we want to publish and that had not been identified in the user requirement phase. Now, we are carrying out the second phase of the process.

Figure 3 shows the refined conceptual model in the UML class diagram notation. The research group is composed by teachers, students and some external collaborators. All members share some information and they also have specific characteristics. The diagram shows the information about the subject given by the teachers in both, degree and Ph.D courses. We want also to maintain information about the students final work, mandatory to get the degree.

Figure 4 shows the slice conceptual model in an adapted RMM notation. In this notation, each slide represents a piece of information, as it should be shown in the Web. This information is taken from different classes as you can see in the example. Slices *Member Student* and *Member Collaborator* contain the information of the class *Member* but *Member Teacher* slice has the information of the class *Member* adding the information of the class *Teacher* except the attribute *Timetable* of the second class. The slice *Final Work* corresponds to the class *Final Work*. The slice *Subject PhD* contains the information of the classes *Subject* and *PhD*. Finally, the slice *Subject Degree* includes the information of the classes *Subject*, *Degree* and the attribute *Timetable* of the class *Teacher*.

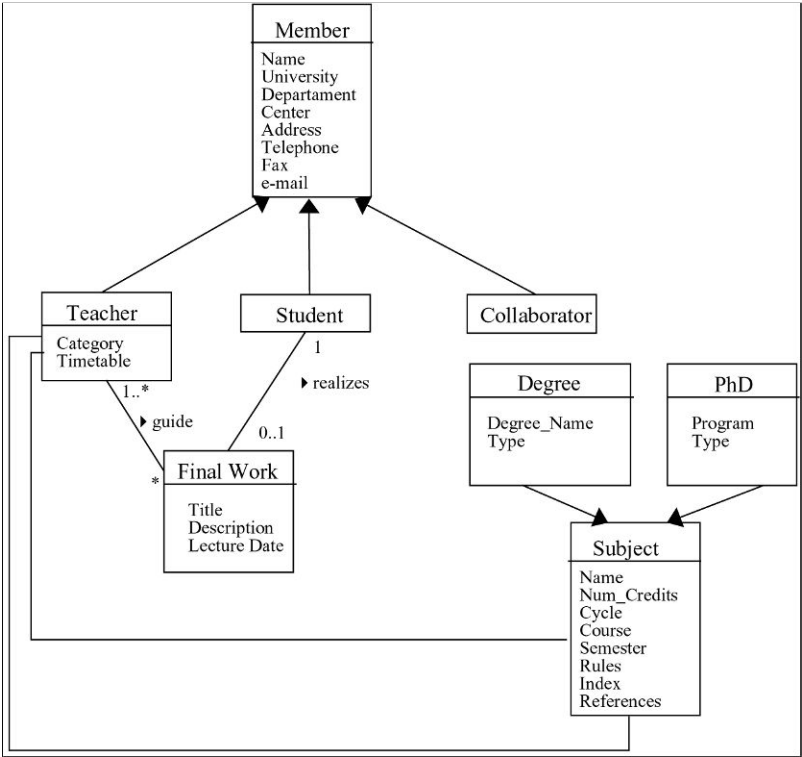


Fig. 3. Refined data conceptual model

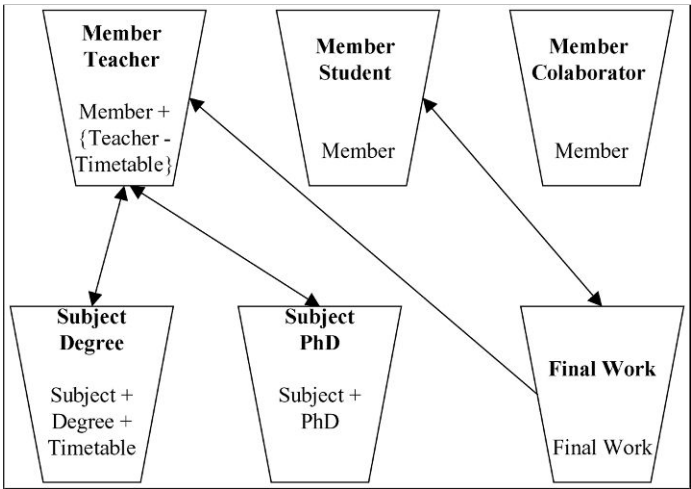


Fig. 4. Refined slice conceptual model

Finally, figure 5 shows the navigational model. Each slice of figure 4 gives rise to a Web page. Besides, menu pages are included and the way of navigating from one page to another (by an index, a direct link, and so forth) is defined.

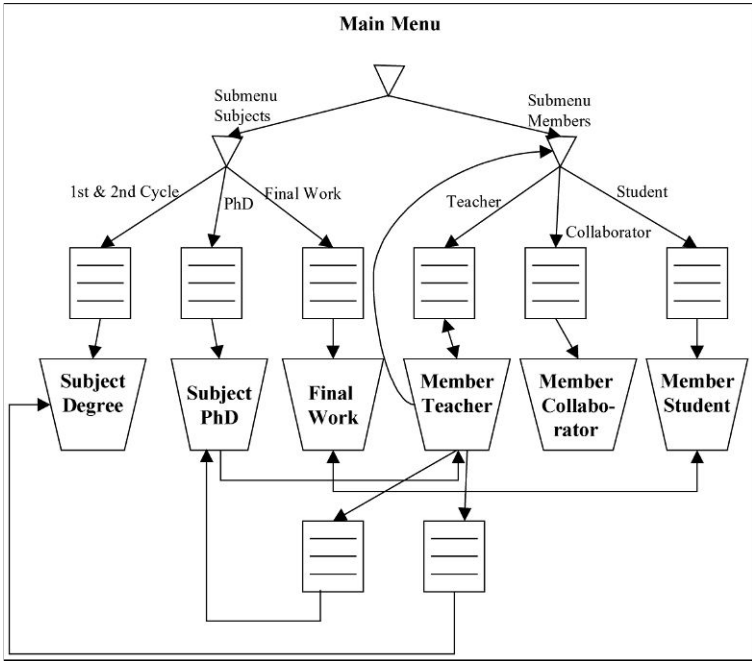


Fig. 5. Refined navigation conceptual model

5 Conclusions and Future Work

MIDAS is a generic methodology that can be customized according to the type of WIS that we want to develop. The methodology proposes an agile process in order to get product versions as soon as possible. Products can be flexibly provided to describe new requirements and so forth in each iteration. In this paper we have explained MIDAS/BD, the specific part of MIDAS for Web database design. It is based on XML and object-relational technology, although it can be easily adapted for either relational or pure object-oriented databases. We have also presented a Web database application developed with MIDAS/BD.

As we have said before, in our opinion it should be better to have a unique notation that allows modeling the whole system. However, although there are some extensions to UML for Web modeling, they do not support all techniques that we have proposed to use in MIDAS/BD. We have already defined an UML extension for object-relational database design [18]. To pass from a step to other in the process it is convenient to provide some guidelines to the developer. In MIDAS/BD we have already defined the guidelines needed to pass from the fifth to the seventh step and from the seventh to the eighth step [17]. Future work includes: a) extending UML to

support each technique proposed in MIDAS/BD; b) defining guidelines to pass from a step to another; c) extending the framework for relational and pure object-oriented database.

Acknowledgements. This work is partially supported by the MIDAS project (CICYT and European Union: 2FD97-2163) and by the HERA project in collaboration with INTESYS (Comunidad de Madrid: 09/0194/2000).

References

1. Atzeni, P; Mecca, G and Meriardo, P; (1998) Design and Maintenance of Data-Intensive Web Sites. Atzeni, Mecca and Meriardo. *Advances in Database Technology*. Ed. Sheck, Saltor, Ramos, Alonso. Proceedings of the 6th. Conference on Extended Database Technology (EDBT'98). Springer Verlag, Valencia.
2. Beck, K. (1999). Embracing Change with eXtreme Programming. *Computer*, vol. 32, nº 10, pp.70-77.
3. Bonifati, A., Stefano, C., Fraternali, P. and Maurino, A.. (2000). Building Multi-device, Content-Centric Applications Using WebML and the W3I3 Tool Suite. *Conceptual Modeling for E-Business and the Web*. Ed. S. W. Liddle, H. C. Mayr, B. Thalheim, Springer Verlag, 2000, pp. 64-75.
4. Cáceres, P. and Marcos, E. (2001). Las Metodologías de Desarrollo y la Mejora de Calidad en las Aplicaciones Web. *Actas del 4º Encontro para a Qualidade nas Tecnologias de Informação e Comunicações*. Lisboa, Portugal, pp. 3-9, Marzo.
5. Castano, S., Palopoli, L. and Torlone, R. (2000) A General Methodological Framework for the Development of Web-Based Information Systems. *Conceptual Modelling for E-Business and the Web*. Ed. S. W. Liddle, H. C. Mayr and B. Thalheim. Springer-Verlag, *Lecture Notes in Computer Science*, Berlin.
6. Conallen, J. (1999). Modeling Web Applications Architectures with UML. *Communications of the ACM*, 42, pp. 63-70. Oct-1999.
7. Conallen, J. (2000). *Building Web Applications with UML*. Addison Wesley.
8. Fournier (1999), *A Methodology for Client/Server and Web Application Development*. Roger Fournier, Prentice Hall.
9. Fowler, M. (2001). *The New Methodology*. Retrieved May 2001 from
10. <http://www.martinfowler.com/articles/newMethodology.html>.
11. Fraternali, P. and Paolini, P; (1998) A conceptual Model and a Tool Environment for Developing more Scalable, Dynamic and Customizable Web Applications. *Advances in Database Technology*. Ed. Sheck, Saltor, Ramos, Alonso. Proceedings of the 6th. Conference on Extended Database Technology (EDBT'98). Springer-Verlag, *Lecture Notes in Computer Science*, Valencia.
12. Fraternali, P. (1999). Tools and approaches for developing data-intensive Web applications: a survey. *ACM Computing Surveys*, Vol. 31, nº 3.
13. Gómez, J. Cachero, C. and Pastor, O. (2000). Extending a Conceptual Modelling Approach to Web Application Design. *Proc. Of the 12th, International Conference on Advanced Information Systems (CaiSE'00)*. Springer-Verlag. *Lecture Notes in Computer Science*.
14. Isakowitz et al. (1995), RMM: A Methodology for Structured Hipermedia Design. *Isakowitz, Stohr y Balasubramanian. Communications ACM*, 38(8), pp. 34-43, August.
15. Isakowitz, T., Kamis, A. and Koufaris, M. (1998). *The Extended RMM Methodology for Web Publishing*. Working Paper IS-98-18, Center for Research on Information System, 1998. Retrieved 23-07-2001 from <http://rmm-java.stern.nyu.edu/rmm/>.

16. Jacobson, Booch and Rumbaugh. The Unified Software Development Process. Addison Wesley, 1999.
17. Lowe and Hall (1999), Hipermedia & the Web. An Engineering Approach. J. Wileyand Sons, 1999.
18. Marcos, E. Vela, B. and Cavero, J.M. (2001). Extending UML for Object-Relational Database Design. UML 2001 – The Unified Modeling Language. Springer Verlag. Lectures Notes in Computer Science. Toronto (Canada), October.
19. Marcos, E. Vela, B. and Cavero, J.M. (2001). Aggregation and composition in object-relational database design. Advanced Databases and Information Systems. Communications of the 5 th East-European Conference ADBIS' 2001. Vilnius (Lituania), September.
20. Mecca, G., Merialdo, P., Atzeni, P., and Crescenzi, V. March (1999). The ARANEUS guide to Web-site development. <http://poincare.inf.uniroma3.it>
21. Schawe and Rossi (1995), The Object-Oriented Hipermedia Design Model. Communications ACM, 58(8), pp. 45-46, August.
22. Schwabe, D. and Rossi, G. (1998) An object oriented approach to web-based applications design. Theory and practice of object system, 4 (4), pp. 207-225.

Translating XQuery into XSLT

Stephan Lechner, Günter Preuner, and Michael Schrefl

Department of Business Information Systems, University of Linz, Austria
{lechner | preuner | schrefl}@dke.uni-linz.ac.at

Abstract. The WWW Consortium (W3C) has recently presented a working draft of XQuery, which is intended to serve as standardized query language for XML. XQuery and other high-level query languages for XML documents are not yet implemented by commercial products. Yet many browsers have already built-in XSLT support for transforming XML documents. XSLT is a standard way of performing structural rearrangement or presentational transformation of XML documents, but formulating complex queries is, compared to XQuery, difficult and error-prone. If XQuery expressions could be translated into XSLT (e.g. by a translator written in Java or XSLT itself), the benefits of XQuery would be immediately available to a wide range of commercial products. This paper introduces a process for translating queries formulated in XQuery syntax into XSL stylesheets. The process is described independently from a particular implementation by means of an ASM (Abstract State Machine). The ASM traverses the parse tree of a particular query and translates each node into corresponding XSLT commands. The result of this translation process is an XSL stylesheet that can be applied to an XML document in order to perform the given query. The presented ASM can be easily coded in Java or XSLT to implement a prototype XQ2XSL translator.

1 Introduction

XML (Extensible Markup Language) [6] has been established by the World Wide Web Consortium (W3C) as a format for data exchange on the web. Typically, objects stored in XML documents obey to a particular structure like objects in databases follow a schema. With the widespread use of XML and the increasing size of XML documents, the need of techniques for comfortably querying XML documents and presenting data selected from these documents has arisen.

Expressing such queries requires techniques to access XML documents and filter the required information from them. One well-known language is XSLT (Extensible Stylesheet Language: Transformations) [9], which has been originally introduced for the purpose of transforming XML documents into a target language for presentation, e.g., into HTML for displaying information in web browsers. Web servers provide XML documents with data of interest together with several XSL stylesheets, each stylesheet defining a particular form of presentation of the data. Web browsers can load and cache XML documents and read the attached XSL stylesheets, thereby giving the user the possibility to

display the same information in different ways without the need to repeatedly reload all data from the web server. Although XSLT is dedicated to define the presentation of data, it could be used for querying, as well.

Yet it turned out that formulating queries in XSLT is a too difficult and error-prone task, raising the need for a high-level query language that is comprehensible to human users and more comfortable to use. Several query languages have been introduced in the past [7,8,10,11,12,16,19]. XQuery seems to be the most promising approach since it has been seized by the W3C, involving researchers who contributed to other query languages before.

XQuery is much more feasible for formulating complex queries than XSLT since it is strongly typed, provides built-in functions, and supports the definition of user-defined functions. User-defined functions can be used to express complex query operations like grouping or joins. Different from that, defining equivalent queries in XSLT is a very complex task. Yet while XSLT has been leveraged by software industry in that XSLT engines have been included into web browsers, query languages for XML have not yet received much attention so far. One reason may be that no standards for query languages have been established yet.

In this paper, we propose a query translator that combines the advantages of a high-level query language like XQuery with the widespread availability of XSLT engines. First, complex queries to access XML documents are defined once using XQuery. The queries are translated to XSLT by the query translator. Finally, further XSLT statements that handle pure presentation aspects (like visual representation of the query results as tables or lists) can be merged into the XSL stylesheets. The resulting XSL stylesheets can be provided to general audience who load the XML documents and XSL stylesheets to browse the data.

Following this idea, applying queries to XML documents involves two subsequent tasks: First, queries written using XQuery are translated into XSL stylesheets (*query translation*); second, XML documents are restructured by applying the XSL stylesheets on them (*query execution*).

Translating queries requires to overcome syntactical and semantical differences between XQuery and XSLT. These differences are caused by the languages' different purposes: While XSLT has been constructed with the purpose of handling recursive and heterogeneous XML data in mind, XQuery is dedicated to complex information retrieval and reuse of intermediate query results [3]. Although several kinds of differences are easily to overcome, some functionality of XQuery cannot be mapped to XSLT in a straight-forward manner. As an example, the result of any expression can be used as input for another expression in XQuery, whereas the result of several kinds of expressions in XSLT must be stored in a variable in order to use it in another expression. This requires to introduce auxiliary XSLT variables during translation.

Translating a query comprises two steps: First, a parse tree is constructed for the query and enriched with information to make the remaining second step easier. Second, the parse tree is traversed in depth-first order, translating each node of the tree into XSL statements by means of *transition rules*.

Translation is presented in an implementation-independent manner by the use of abstract state machines (ASM). ASMs were originally introduced as *evolving algebra* [15] and were successfully applied for the specification of the semantics of programming languages [4,5,21] and object-oriented data models (e.g., [14,20]) in the past. Compilers for translating ASMs can be used to implement the XQuery translator (cf., e.g., [2] for a compiler of ASMs to programming language C).

This work focuses on the process of translation; it factors out accompanying tasks like static type checking of input queries and leaves the use of XQuery runtime type information to future work.

The remainder of this paper is organized as follows: In Sect. 2, we explain differences between XQuery and XSLT. Section 3 gives an overview of the overall process of query translation and query execution. In Sect. 4, the translation process is presented in detail by means of abstract state machines. Finally, Sect. 5 concludes the work.

2 XQuery versus XSLT: Comparison

This section compares XQuery with XSLT. Concepts with no straight-forward correspondence are identified along two dimensions, differences between data models and between language constructs, and are illustrated by means of a running example based on an example in the XQuery working draft [13]. Given is an XML-document that describes a collection of books (cf. Fig. 1) and contains a comment and a processing instruction. The sample XQuery depicted in Fig. 2 selects from this document the titles of books published in a particular year. The query uses a function `annualBooks`, defined within the query, to select the desired book elements before the titles are retrieved. The query's result is shown after the `==>`-sign. The query assumes that a global variable `root`, which is not shown in Fig. 2, holds the book collection.

Differences between Data Models. Though XQuery and XSLT take XML documents as input, they internally work on different data models that provide different information. This gives rise to various problems due to information used by XQuery not being available in XSLT.

(1) *Node Identity:* In XQuery, each element, attribute, comment, and processing instruction (referred to commonly as nodes) has an identity. Except for expressions used within an element constructor, XQuery preserves node identity such that the result of two XQuery expressions may contain identical nodes. The XQuery node identity operator (`==`) can be used to check whether two nodes are the same, as illustrated in Fig. 3. XSLT, in contrast, generates new identities each time an expression's result is copied into an XSLT variable such that the identity of an node is lost when an interim result is stored temporarily in an XSLT variable.

(2) *Incompatible Domains of Interim Query Results:* XQuery allows to reuse all sorts of interim query results. For example, the result of projecting on at-

```

<bib>
  <!-- some comment -->
  <? someTarget instructions ?>
  <book year="2001" isbn="1-XXXXX-YYY-Z">
    <title>XML Query</title>
    <author>Fernandez</author>
    <author>Suciu</author>
  </book>
  <book year="1999" isbn="1-55860-622-X">
    <title>Data on the Web</title>
    <author>Abiteboul</author>
    <author>Buneman</author>
    <author>Suciu</author>
  </book>
</bib>

```

Fig. 1. Segment of an XML document describing a collection of books

```

DEFINE FUNCTION annualBooks
(xs:integer yearParam)RETURNS Book* {
  FOR $x in $root/book
    WHERE $x/@year/data() = $yearParam
    RETURN $x
}

FOR $b in annualBooks (2001) RETURN
  LET $b = $b/title RETURN
  $b

==> <title>XML Query</title>

```

Fig. 2. XQuery selecting recently published books

```

LET $b1 = $book0/title,
    $b2 = $book0/title RETURN
  $b1 == $b2
==> TRUE

LET $b1 = <newBook>{$book0/title}</newBook>,
    LET $b2 = <newBook>{$book0/title}</newBook> RETURN
  $b1/title == $b2/title
==> FALSE

```

Fig. 3. Applying XQuery's node identity operator

tributes (i.e. a set of attributes) can be stored in a local variable, despite being not a valid XML document, and can be used for further processing. XSLT, in contrast, treats variables as output if the value is defined within the variable definition's body. Therefore, copying a set of attributes into an XSLT variable would lead to an error, even if only used as interim query result.

Differences between Language Constructs. The following features of XQuery are not available in XSLT:

(1) *Closure*: The result of an XQuery expression may be direct input to other expressions. This feature is referred to as *closure*. E.g., the query shown in Fig. 2 iterates over a set of books achieved from a function call and performs the inner expression for each book. The result of some XSLT expressions may be not used as direct input to other expressions. E.g., passing the result of a template call to another XSLT expression requires to store the interim result in an auxiliary variable which is used as input to the next expression.

(2) *Nested Variable's Scopes*: XQuery permits nested scopes of local variables, i.e. a local variable named v can be defined within the scope of another local variable named v . If an expression refers to a variable, the value of the most recent variable definition is bound. For example, the variable $\$b$ defined by the FOR-expression in Fig. 2 is redefined by another local variable named $\$b$. XSLT, in contrast, does not allow nested scopes of local variables.

(3) *Parameter Binding by Position vs. Parameter Name*: In XQuery, actual parameter values in a function call are bound to the formal parameters by

their position. For example, when calling the function `annualBooks (2001)` in Fig. 2, the first actual parameter (2001) is bound to the first formal parameter (`yearParam`). Functions in XQuery correspond to named templates in XSLT, but when calling a template, each actual parameter value must be embedded in an `<xsl:with-param>`-element naming the formal parameter to be bound.

(4) *Reference and Dereference Operators*: With the reference operator (`ref(expression)`), XQuery allows to store a reference to the result of an expression. The identity of each node in the expression's result is preserved, even if the reference operator is used within an element constructor. The dereference operator (`deref(expression)`), as the reference-operator's counterpart, re-constructs the referred expression's result. Fig. 4 illustrates an application of the reference- and the dereference operators. XSLT does not define generally applicable reference- and dereference operators. In particular, there is no way to generate a reference to a node of an arbitrary interim query result so that this reference can be used in an expression anywhere else to retrieve the same node that the reference was generated for.

```
LET $b1 = <newBook> { ref ($book0/title) } </newBook>
RETURN
  $book0/title == deref ($b1/*)

==> TRUE
```

Fig. 4. Reference and dereference operator

3 Overview of XQuery Translation and Execution

In this section, the process of performing XQueries on XML documents is introduced, together with strategies on how to resolve differences between XQuery and XSLT as discussed in Sect. 2.

The process comprises two main steps (cf. Fig. 5 for an overview): First, the query translator translates the XQuery to an XSL stylesheet (*query translation*); second the XSLT processor applies the XSL stylesheet to the XML document (*query execution*). Differences between internal data models are handled by pre-processing (Step 2a) each XML document before applying the translated query to it (Step 2b). The query result is post-processed in that auxiliary information produced by the preceding steps is removed (Step 2c).

Since the complete process of query execution is performed by an XSLT processor, all three steps of query execution must be controlled by one XSL stylesheet, which is generated by the query translator. The XSL stylesheet comprises three logical parts, one for each step of query execution.

For the purpose of this overview of translation and execution, the whole process of evaluating XQueries on XML data (comprising both translation and

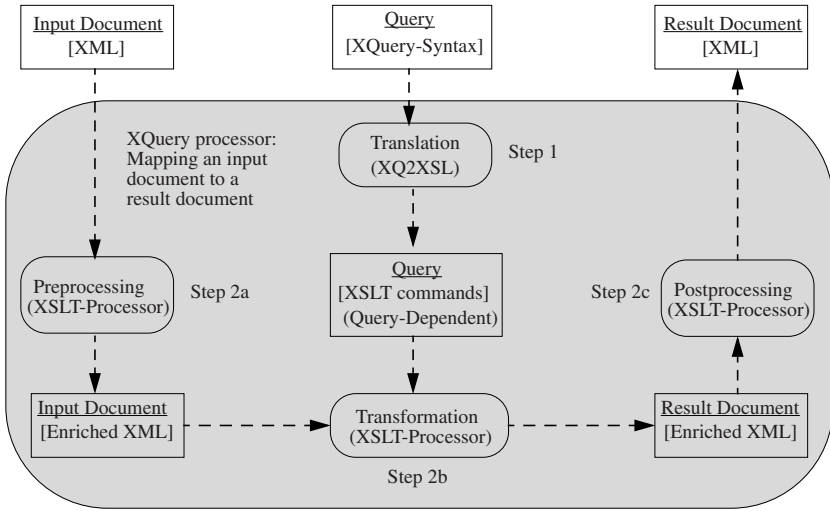


Fig. 5. XQuery translation and execution

execution) is explained along the three steps of query execution. For each step, the changes on XML data, managed by XSLT commands, are explained; further, the way how differences by XQuery and XSLT can be overcome is illustrated.

3.1 Preprocessing

In this step, the input XML document is enriched by auxiliary information to manage *data model differences*. Those parts of the XSL stylesheet that control preprocessing are *query-independent*, i.e., are the same for all queries.

The problem of *incompatible domains of interim query results* is resolved by wrapping attributes, comments, and processing instructions by elements.

Node identity, which is not preserved by XSLT in general, is ensured by introducing an identifier attribute for each element such that identity becomes explicit and is not lost during copying (cf. Sect. 2).

Figure 6 shows the enriched representation for our sample XML document given in Fig. 1. Items added to the input document are identified by a namespace-prefix `idm` (standing for “internal data model”). The input document’s attributes, comments, and processing instructions are wrapped by special *wrapper elements* tagged with `idm:ATTRIBUTE`, `idm:COMMENT`, and `idm:PI`, respectively. Each element is extended by an identifier attribute, `idm:id-attribute`, whose value is unique within a document.

3.2 Applying the Query

The actual query is applied to the enriched document by the *query-dependent* part of the XSL stylesheet. The relevant XSL statements are defined by the


```

<bib idm:id="1">
  <idm:COMMENT idm:id="2">
    some comment text
  </idm:COMMENT>
  <idm:PI idm:id="3" target="someTarget">
    instructions
  </idm:PI>
  <book idm:id="4">
    <idm:ATTRIBUTE idm:id="5" year="">
      2001
    </idm:ATTRIBUTE>
    <idm:ATTRIBUTE idm:id="6" isbn="">
      1-XXXX-YYY-Z
    </idm:ATTRIBUTE>
    <title idm:id="7">XML Query</title>
    <author idm:id="8">Fernandez</author>
    <author idm:id="9">Suciu</author>
  </book>
  <book idm:id="10">
    <idm:ATTRIBUTE idm:id="11" year="">
      1999
    </idm:ATTRIBUTE>
    <idm:ATTRIBUTE idm:id="12" isbn="">
      1-55860-622-X
    </idm:ATTRIBUTE>
    <title idm:id="13">Data on the Web</title>
    <author idm:id="14">Abiteboul</author>
    <author idm:id="15">Buneman</author>
    <author idm:id="16">Suciu</author>
  </book>

```

Fig. 6. Enriched XML document of Fig. 1

```

<xsl:template name="annualBooks">
  <xsl:param name="yearParam"/>
  <xsl:copy-of select=
    "$root/book[idm:ATTRIBUTE[@year]/text()
    =$yearParam]"
  />
</xsl:template>

<xsl:template match="/">
  <xsl:variable name="FOR1_outer">
    <xsl:call-template name="annualBooks">
      <xsl:with-param name="yearParam"
        select="'2001'" />
    </xsl:call-template>
  </xsl:variable>
  <xsl:for-each select="$FOR1_outer/*">
    <xsl:variable name="FOR1"
      select="."/>
    <xsl:variable name="LET1">
      <xsl:copy-of select="$FOR1/title"/>
    </xsl:variable>
    <xsl:copy-of select="$LET1"/>
  </xsl:for-each>
</xsl:template>

```

Fig. 7. XSL stylesheet fragment corresponding to the XQuery of Fig. 2

```

<title idm:id="7">XML Query</title>

```

Fig. 8. Result in enriched format

query translator during query translation, which comprises the following two steps: First, a parse-tree of an XQuery is generated and augmented by auxiliary information, such that its later processing is simpler; second, the parse tree is traversed and each node is translated to XSL statements. Details of this step will be presented in the next section. The differences between language constructs identified in Sect. 2 are coped with as follows:

(1) *Closure*. If the result of an XSLT expression cannot be directly processed by subsequent expressions, an auxiliary variable is introduced for storing the interim result and passing it to other expressions. E.g., in the stylesheet of our running example (cf. Fig. 7) the result of calling template `annualBooks` is stored in an auxiliary XSLT variable, `FOR1_outer`, which is input for the `<xsl:for-each>`-statement.

(2) *Nested variable's scopes*. Nested local variable definitions in an XQuery would cause naming conflicts if they were mapped to local XSLT variables using the same names as in the XQuery. To avoid this, each local variable defined in an XQuery gets a different name in the corresponding stylesheet. The name for a local XSLT variable representing a particular XQuery local variable definition v is called the *alias* of v . If an XSLT expression refers to a particular variable, the

proper alias is used for binding a variable according to XQuery's scope resolution. For example, the two local variables of Fig. 2 named `b` are represented by two local variables named `FOR1` and `LET1` in the stylesheet of Fig. 7.

(3) *Parameter binding by position vs. parameter name.* The names of formal parameters in XQuery function definitions are used in XSLT template calls to which an XQuery function call is translated (cf. `<xsl:call-template>` in Fig. 7).

(4) *Reference and dereference operators.* Though XSLT does not define a general applicable way to reference arbitrary nodes, reference and dereference can be simulated by storing an identity preserving copy of the referenced items as a reference value. This copy is wrapped by an internal element `idm:ref` in order to distinguish reference items from other items.

Consider once more the XQuery shown in Fig. 2: The query-dependent XSL statements are depicted in Fig. 7. The result of applying the query to the enriched XML document in Fig. 6 is presented in Fig. 8.

3.3 Postprocessing

Auxiliary elements, which were produced by the previous step of applying the query, are removed during postprocessing. In the example, postprocessing of the XML document depicted in Fig. 8 yields the result shown in Fig. 2.

4 XQ2XSL: Query Translation Process

In this section, we describe how to translate an XQuery into an XSL stylesheet. The general approach is as follows: (1) the query is represented by its parse tree which is augmented with auxiliary information such as aliases for local variables to simplify the subsequent translation process, and (2) the parse tree is traversed and the nodes are translated into corresponding XSLT statements.

To be independent from particular implementation languages, we use *abstract state machines* [15] to define the translator's semantics given an augmented parse tree of the XQuery.

An ASM (abstract state machine) consists of a many-sorted, partial algebra (comprising universes and functions) and a set of transition rules. The actual state of an ASM is represented by the state of the algebra, whereas state changes are defined by transition rules. Each transition rule consists of a closed boolean condition over the signature of the algebra and a local function update. A local function update modifies for an n -ary function f of the algebra the image $f(x)$ for a particular x . The local function update of a transition rule is performed if and only if the boolean condition attached with the transition rule evaluates to true. Universes and functions that are modified by transition rules are called *dynamic*, whereas all others are called *static*.

The parse tree of the input XQuery is described by universes and functions of an ASM. The traversal of the tree as well as output is described by transition rules and dynamic functions. In the following subsections, we describe the generation of the parse tree, its traversal and specific transition rules for translating nodes into XSLT instructions.

4.1 Parse Tree Generation

The parse tree of an XQuery is an edge-labeled tree of typed nodes. Nodes represent XQuery expressions, node types the kind of expression. Edge-labels identify the role a child node plays according to XQuery's semantic in the expression denoted by the parent node. Leaf nodes have attached literals (i.e. strings, number constants, or variable names). Figure 9 shows a fragment of the parse tree of the XQuery depicted in Figure 2. Node inscriptions show the node type postfixed by an index to give a unique node identity. Ignore annotations shown in gray underlays for the moment.

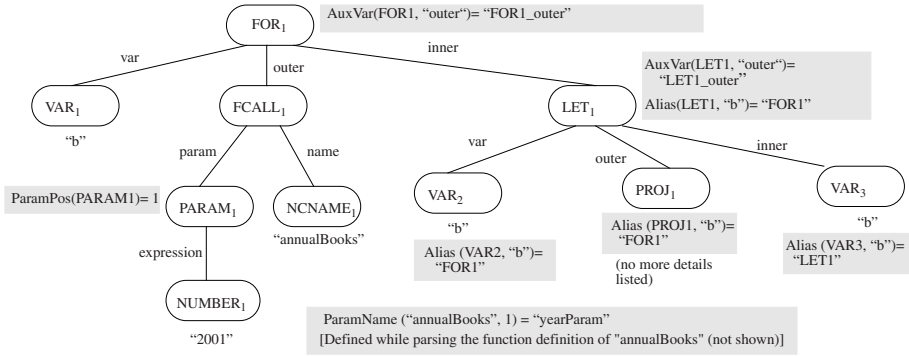


Fig. 9. Parse tree fragment for XQuery of Fig. 2

The parse tree of an XQuery is represented in an ASM by the following universes and functions. Unless stated otherwise, universes are static and functions total and static.

<i>Universe</i>	represents
<i>nodes</i>	the set of nodes of the tree
<i>typeNames</i>	names of all node types
<i>labels</i>	labels of edges from parent to child nodes
<i>strings</i>	node values and literal values (dynamic)
<i>Function</i>	maps a node to
<i>NodeType</i> : $nodes \rightarrow typeNames$	its node's type
<i>Child</i> : $nodes \times labels \rightarrow nodes$	the child node identified by the label (partial)
<i>Parent</i> : $nodes \rightarrow nodes$	the node's parent (partial)
<i>LiteralValue</i> : $nodes \rightarrow strings$	a node's literal value (partial)

The strategies to overcome language differences introduced in the previous section require the introduction of auxiliary variables, the definition of aliases for variables, and the mapping of positional function parameters to parameter names. These tasks can be easily performed by the parser during the generation

of the parse tree, since they are specific to the nodes parsed. The parse tree is augmented as follows: If the parser recognises an expression that requires to store interim or final results, it generates auxiliary variables for these and annotates them to the node expression parsed. If the parser detects a variable definition, it introduces an alias for the variable and attaches this alias to all nodes in the scope of the variable definition such that the translator needs not to traverse the parse tree when resolving variable names. If the parser parses a function definition, it establishes a mapping between positions and parameter names; if the parser parses a function call, it attaches a position number to each parameter. Generated variable and parameter names are kept unique by including the node-id at which they are introduced in their names. In the ASM representation, the functions listed below capture these augmentations of the parse tree. The gray-shaded annotations of Fig. 9 show values of these functions as they are set while parsing the XQuery of Fig. 2.

<i>Functions of augmented parse tree</i>	function call : result
$AuxVar : nodes \times labels \rightarrow strings$	$AuxVar(n, l)$: auxiliary variable at node n for interim result of child l
$Alias : nodes \times strings \rightarrow strings$	$Alias(n, s)$: alias for XQuery-variable s at node n
$ParamName : strings \times integer \rightarrow strings$	$ParamName(f, i)$: name of the i -th formal parameter of function f
$ParamPos : nodes \rightarrow integer$	$ParamPos(n)$: position of parameter node n (defined for parameter nodes only)

4.2 Parse Tree Translation

The enriched parse tree is translated to an XSL stylesheet in that XSL statements are defined for each node of the tree.

The actual state of the tree is represented by the ASM A_{xq2xsl} . Changes of the parse tree during translation are reflected by two dynamic functions:

Function NV : $nodes \rightarrow strings \cup \{“uneval”\}$ maps each node to a string of XSL statements. Initially, each node n is unevaluated, i.e., $NV(n) = “uneval”$. If a node is evaluated, $NV(n)$ is changed from “uneval” to an element of *strings*.

0-ary function AN points to the active node, i.e., the node currently evaluated. Initially, AN resides at the root of the parse tree.

The tree is traversed in depth-first order. The child nodes of a particular node are traversed from left to right (although any other ordering would be permissive, as well). Each node is evaluated only if all of its child nodes have already been evaluated. *Transition rules* are defined for tree traversal (specifying a depth first processing) and for node translation (as presented in the remainder of this section (cf. Fig. 10)).

```

if NV(AN)="uneval" and
   $\nexists c \in \text{Children}(AN) : \text{NV}(c) = \text{"uneval"}$  and
  NodeType(AN)="ForExpression" then
  NV(AN) :=
    <xsl:variable name="%1">%2</xsl:variable>
    <xsl:for-each select="%$1/*">
      <xsl:variable name="%3" select="."/>
      %4
    </xsl:for-each>
endif

```

```

NV(AN)=
  <xsl:with-param name="%1">%2</xsl:with-
  param>

  let parent=Parent (AN) in
  let fname=LiteralValue(Child (parent, "name")) in
  %1= ParamName (fname, ParamPos (AN))
endlet endlet

```

(b) Body of transition rule for a function call parameter

```

%1=AuxVar(AN, "outer"),
%2=NV(Child(AN, "outer")),
%3=Alias(Child(AN, "inner"),
  LiteralValue (Child (AN, "var"))),
%4=NV(Child(AN,"inner"))

```

```

NV(AN)=
  ref (Expr:expression) <idm:ref>%1</idm:ref>

  NodeType (AN) = "ReferenceOp"
  Child (AN, "expression")

```

(a) Transition rule for a FOR-expression

(c) Node and body of transition rule for reference operator

```

deref (Expr:expression)
  NodeType (AN) = "DereferenceOp"
  Child (AN, "expression")
  AuxVar (AN, "expression") =
    "DEREF1_expression"

```

```

NV(AN)=
  <xsl:variable name="%1">%2</xsl:variable>
  <xsl:copy-of select="%$1/idm:ref"/>

  %1= AuxVar (AN, "expression")
  %2= Child (AN, "expression")

```

(d) Node and body of transition rule for dereference operator

```

Expr:LHSExp == Expr:RHSExp

  NodeType (AN) = "NodeIdentityOP"
  Child (AN, "LHSExp")
  Child (AN, "RHSExp")
  AuxVar (AN, "RHS") =
    "IDOP1_RHS"
  AuxVar (AN, "LHS") =
    "IDOP1_LHS"

```

```

NV(AN)=
  <xsl:variable name="%1">%2</xsl:variable>
  <xsl:variable name="%3">%4</xsl:variable>
  <xsl:value-of select="%$1/@idm:id = $%3/
  @idm:id"/>

  %1= AuxVar (AN, "LHS")
  %2= Child (AN, "LHSExp")
  %3= AuxVar (AN, "RHS")
  %4= Child (AN, "RHSExp")

```

(e) Node and body of transition rule for node identity operator

```

attribute QName:name (
  Exp:expression
)

  NodeType (a) = "AttributeConstructor"
  Child (a, "name")
  Child (a, "expression")

```

```

NV(AN) =
  <idm:ATTRIBUTE
  idm:id="{generate-id()}"
  >
  <xsl:attribute name="%1"/>
  %2
  </idm:ATTRIBUTE>

  %1= NV (Child(AN, "name"))
  %2= NV (Child(AN, "expression"))

```

(f) Grammar, parse tree node and body of transition rule for attribute constructor

Fig. 10. Transition rules for translation

4.3 Transition Rules for Selected Parse Tree Nodes

For every type of node, a transition rule is defined for translating the node, i.e. assigning the node's value an XSLT statement, once its children have been evaluated. Due to space limitations, we can present transition rules only for selected parse tree nodes. The rules selected, however, cover the range of differences identified in Sect. 2. Since the conditions of these rules follow always the same pattern, we will present the rule condition only for one rule.

The bodies of these rules always evaluate to an XSLT-string that consists of a node-dependent constant string interspersed with information compiled from child node values. For better readability, we embed place holders denoted by %_{*i*} in XSLT strings and show them expanded separately (cf. Fig. 10 (a)).

(1) *Transition rule coping with closure*: The translation of an XQuery FOR-expression iterates over the result of its outer expression. A variable for holding this result has already been prepared when the parse tree was augmented. It is now used to provide the input for the <xsl:for-each>-statement as illustrated in Fig. 10 (a).

(2) *Transition rule coping with nested variable's scopes*: Globally unique alias names have been introduced during parse tree augmentation for all local variables and have been associated with each node in a variable's scope. These alias names can now be used when generating XSLT statements. E.g., when the transition rule for a FOR-expression is applied to node FOR1 of the parse tree shown in Fig. 9, the alias name FOR1 will be used for the loop-variable corresponding to **b** (cf. the use of the *Alias*-function associated with place holder %₃ in Fig. 10 (a)).

(3) *Transition rule dealing with parameter passing by name*: When calling a template that represents an XQuery function, each actual parameter must be translated into an <xsl:with-param>-element including the name of the corresponding formal parameter. This name is determined by first using the ASM *ParamPos*-function to retrieve the parameter's position and by then using this position to determine the formal parameter's name by ASM *ParamName*-function. Figure 10 (b) depicts the body of the transition rule for translating a parameter of a function call.

(4) *Transition rules coping with reference and dereference*: As explained above, these operators are simulated by copying. A reference operator's result is an identity-preserving copy of the referenced items wrapped by an **idm:ref**-element. The dereference-operator simply un-wraps these items. Figure 10 (c) and Fig. 10 (d) illustrate the bodies of transition rules for the reference- and dereference-operator, respectively.

(5) *Transition rules coping with node identity*: Node identity, as queried by the expressions shown in Fig. 3, is checked by the node identity operator (==) and implemented using the **idm:id**-attribute available in the enriched XML document representation.

The node identity operator compares the values of its operand's **idm:id**-attributes. Figure 10 (e) shows the grammar, functions representing the corresponding parse tree node, and the body of the transition rule for this operator.

The `idm:id`-attribute of elements simulates the semantics of XQuery's node identity when a document is processed by an XSLT processor. Consequently, the `idm:id`-attribute will be identical to the internal node identity generated by an XSL processor, unless an element is copied outside an XQuery element constructor. Then the copied element will have a new XSL identity but keeps its old `idm:id`-identity.

Therefore, outside XQuery element constructors, the `xsl:copy-of`-function can be used as identity-preserving copy function, since every `idm:id`-attribute is copied as is. The copy-function for XQuery expressions within element constructors needs to be translated into a deep copy, whereby the copied elements receive a new `idm:id`-attributes value. For sake of space, we do not show the transition rule for XQuery's element constructor. But the setting of an `idm:id`-attribute is shown below for the XQuery attribute constructor which requires the generation of a wrapper element with an `idm:id`-attribute.

(6) *Transition rules dealing with incompatible domains of interim query results:* Since it is not possible to copy attributes into XSLT variables, all attributes are wrapped by `idm:ATTRIBUTE`-elements in the enriched XML document representation. Consequently, when generating a new attribute, these wrapper elements are generated, too, and the `idm:id`-attribute of this element is set by using the node identity generated by the XSL processor. Figure 10 (f) illustrates the grammar, a parse tree node and the body of the transition rule for translating an attribute constructor.

5 Conclusion

In this paper, we presented a translation of XML document queries formulated in XQuery to XSLT. Our research is motivated in that high-level query languages like XQuery are comfortable to use, yet software for evaluating XQueries is missing. Different from that, XSLT is a widely distributed language for transforming XML documents for presentation and can be used as a query language, as well; whereas writing queries in XSLT is a difficult and error-prone task.

Our query translator XQ2XSL combines the advantages of XQuery and XSLT in that it translates XQueries into XSL stylesheets, which can be processed with many web browsers or tools that provide XML/XSL support. The resulting XSL stylesheets can be enhanced with statements for presentational markup.

We presented the translator by means of an abstract state machine that takes an enriched parse tree of the XQuery as input and translates each node of the tree to XSL statements.

For our XQ2XSL translator currently under development, we use the parser generator CUP [17], which is a Java-pendant to the YACC parser generator [18] available on UNIX systems, to generate a parser that maps XQueries into object trees. XQuery's grammar is denoted as a *translation scheme*, i.e., a context-free grammar in which program fragments called *semantic actions* (cf. [1]) are embedded within the right-hand side of productions for generating the described augmentations of the parse tree. The ASM is implemented in Java in a straight-

forward manner. The first version of the XQ2XSL translator does not support the translation of XQueries that make use of XQuery's runtime type information. We are currently extending our concept in that direction.

References

1. J. D. Ullman A. V. Aho, R. Sethi. *Compilers: Principles, Techniques and Tools*. World Student Series. Addison Wesley, 1986.
2. M. Anlauff. XASM - An Extensible, Component-Based ASM Language. In *Proc. ASM 2000*. Springer LNCS 1912, 2000.
3. XSL Working Group at the W3C. The Query Language Position Paper of the XSL Working Group (Draft 11/18/98). <http://www.w3.org/>, 1998.
4. E. Börger and D. Rosenzweig. A mathematical definition of full Prolog. In *Science of Computer Programming*. 1994.
5. E. Börger and W. Schulte. A Programmer Friendly Modular Definition of the Semantics of Java. In *Formal Syntax and Semantics of Java*. Springer LNCS 1523, 1998.
6. T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. Extensible Markup Language (XML) 1.0 (Second Edition). <http://www.w3.org/>, October 2000.
7. P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. A Query Language and Optimization Techniques for Unstructured Data. In *Proc. ACM-SIGMOD*, 1996.
8. D. Chamberlin, J. Robie, and D. Florescu. Quilt: An XML Query Language for Heterogeneous Data Sources. In *Proc. WebDB*, 2000.
9. J. Clark. XSL Transformations (XSLT) Version 1.1. <http://www.w3.org/TR/xslt11>, December 2000. Working Draft.
10. S. Cluet, S. Jacqmin, and J. Sim'eon. The New YaTL: Design and Specifications. Technical report, INRIA, 1999.
11. A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. A Query Language for XML. In *Proc. WWW8*, 1999.
12. P. Fankhauser, M. Fernandez, A. Malhotra, M. Rys, J. Simeon, and P. Wadler. XQuery 1.0 Formal Semantics, Working Draft. <http://www.w3.org/>, June 2001.
13. M. Fernández and J. Marsh. XQuery 1.0 and XPath 2.0 Data Model: Working Draft. <http://www.w3.org/>, June 2001.
14. G. Gottlob and M. Schrefl. The Evolving Algebra Semantics of Class and Role Hierarchies. In *Semantics in Databases*. Springer LNCS 1358, 1998.
15. Y. Gurevich. Logic and the challenge of computer science. In *Trends in Theoretical Computer Science*. Computer Science Press, 1988.
16. H. Hosoya and B. Pierce. XDuce: A Types XML Processing Language (Preliminary Report). In *Proc. WebDB*, Springer LNCS 1997, 2000.
17. S. E. Hudson. CUP LALR Parser Generator For Java: User's Manual. <http://www.cs.princeton.edu/~appel/modern/java/CUP/manual.html>, 1999.
18. S. C. Johnson. YACC — Yet Another Compiler Compiler. Technical Report C.S. Rechnical Report #32, AT & T Bell Labs, 1975.
19. J. Robie. XQL '99 Proposal. <http://www.ibiblio.org/xql/xql-proposal.html>, 1999.
20. M. Schrefl and G. Kappel. Cooperation Contracts. In T. J. Teorey, editor, *Proc. 10th Int. Conf. on Entity-Relationship Approach (ER '91)*, 1991.
21. C. Wallace. The Semantics of the C++ Programming Language. In *Specification and Validation Methods*. Oxford University Press, 1995.

Web Site Evaluation: Methodology and Case Study

Paolo Atzeni¹, Paolo Merialdo¹, and Giuseppe Sindoni²

¹ D.I.A. – Università di Roma Tre, via della Vasca Navale, 79, I-00146 – Roma, Italy,

² ISTAT, via Adolfo Ravà, 150, I-00184 – Roma, Italy

[atzeni, merialdo,sindoni]@dia.uniroma3.it

Abstract. This paper presents a methodology for the assessment of Web site quality with a particular emphasis public sector Web sites. In order to rate a site, a hierarchical model is proposed, which comprises several quality attributes. The methodology is designed for use by independent analysts who may have no knowledge of the technology underlying the site nor any contact with the site managers. We used the methodology to assess the quality of several Web sites of the Italian Public Administration. The results of the analysis are presented as a case study.

1 Introduction

The Web is evolving from an initial stage, in which information presentation was the principal goal of sites, to a global network of more and more sophisticated services. This makes Web site quality one of the main issues to be taken into account through all phases of the life-cycle of a given site.

In this paper, we report on an effort we have recently completed under the auspices the Italian Authority for IT in Public Administration (AIPA) for assessing the status of the Web sites of the offices of the Central Government of the country, as a preliminary step to improving their quality. We defined, and then applied, a method for analyzing the sites. The major findings of the research were that quality can be achieved only by means of a disciplined approach that includes specific steps that foster quality and others that allow for a continuous assessment, with the involvement of internal personnel as well as external, independent evaluators.

The issue of Web-site quality is dealt with (often implicitly) in all pieces of work concerning Web development (for example the books by Lynch and Horton [8] and by Rosenfeld and Morville [12]). Usability, a specific aspect of quality, is often considered explicitly, and there is a large body of printed and electronic literature on it (including the books by Nielsen [10] and Spool [14]).

Indeed, Web quality presents a number of different aspects that can be hardly summed up, since most of them are in fact incomparable. For example, how could one say whether it is better to have a fast Web site with obscure navigation paths or one with a very slow access but very intuitive and elegant structure? In the

private sector, there could be a synthetic measure for the quality of a site: the profit (which could in turn be measured by sale revenues for e-commerce sites or number of contacts for sites that get money from advertisers). In the public sector, the goal is to provide “service” (to individual citizens or to businesses), but it is in general impossible to synthesize the quality of the service by means of a single figure: in most cases, the service has to be general, offered to all without discrimination and barriers of any kind. There can be “political choices,” setting priorities, but this usually have little or no technical motivation; instead, they are part of the input requirements.

As a consequence, we found that Web quality could be defined by means of various different features, each to be pursued, often independently of the others. In fact, our assessment method, as we will discuss in the next section, considers about 50 different features, which, for better readability of the results, have been organized in categories and subcategories. In this way, the assessment of the status of a site can provide very precise input on which are the aspects that deserve specific attention.

The existence of many different aspects that influence quality can be confusing, since it could be difficult for site managers to understand how to handle all of them. To this end, we propose a methodological framework for Web development that includes specific steps for considering each of the quality features of our assessment method. Interestingly, we found that many specific features could be verified in a local way, on each page or on specific pages of the site, and therefore they could be taken into account in a rather simple way, by means of checklists. Instead, the most general features (such as the coherence with the goals and the quality of the information architecture), should first of all taken into consideration by means of a principled approach to development (and then also verified “a-posteriori”): they can hardly be corrected, if not adequately set up at the beginning. This is especially true for the overall goals of a site: if the purpose and the intended audience are not clearly specified at the inception, there is no way to get the “right” site. Therefore, the framework includes development activities and quality checking activities, the latter to be performed with the involvement of external subjects.

The proposed method is composed by two distinct analysis phases and models Web quality by means of 46 features, which have been organized in categories and subcategories. The features were collected from those presented in the literature [8,10,12], and selected according to the experience of many site managers, designers and developers. Most deal, either directly or indirectly, with Web site quality. In this way, the assessment of site status can give very precise information on which data aspects deserve specific attention. It is worth noting that our features hierarchy is different from the attribute hierarchy which has been proposed in other works [7] and which was driven from the hierarchy of quality attributes of databases [2]. In fact, one of the goals of our method is to make a clear distinction between “measurable” features, to which a numeric evaluation can be associated, and general features, on which only a qualitative analysis

can be performed. However, most features included in our model can easily be mapped to the Data Quality Dimensions presented by the authors of [2].

The paper is organized as follows. In Section 2 we describe the various features we have included in our assessment model and method. In Section 3 we briefly report on the application of the methodology. Finally, in Section 4 we present some concluding remarks.

2 Quality Features for Web Sites

Our method is based on the general ideas discussed in the previous section and was defined taking into account the following requirements.

First, the method has to be general, and not try to deal with the peculiarities of a specific organization. For example, we did not make experiments on the effectiveness of search tools on site-specific information: this would normally require groups of representative users for the site. However, we report this aspect in the methodological framework, suggesting that site managers perform this activity with the support of user groups as part of their internal quality assessment. Second, since the analysis should be completely independent (we only use information available via the site), the method can be applied even without interacting with the site managers. Finally, since the method aims at a “horizontal analysis” - i.e. the relative comparison among different sites of the same business area - objective, measurable aspects are emphasized and the method is scalable with respect to the number of sites involved.

We organized the site analysis in two phases: *General Analysis* and *Specific Analysis*. The former focuses on aspects that can be evaluated mainly on a qualitative basis; the latter considers features on which a quantitative evaluation can be performed.

It should be noted that, given the above requirements, in our method the quantitative aspects have a higher relative importance than the qualitative ones.

2.1 General Analysis

The general analysis deals with the overall features of the site and aims at providing a succinct yet complete snapshot of the site. We have identified four main features that help build the picture: *Objectives*, *Structure*, *Services*, and *Effectiveness*.

Site Objectives. This is the answer to the strategic question “*What is the site for?*”. Knowledge of the objectives is fundamental for the data evaluation process, and must be acquired at the very beginning of the study. Some sites explicitly declare their goals; for others, if an interview with the administration people is not possible, the goals can be deduced from the site’s content. The site objectives can be divided into two main categories: (i) generic, which are independent from the peculiarities of the individual administration (for example, providing information about the internal organization or

publishing related laws and regulations); (ii) domain-specific, which refer to services or information of interest to the users of that site (for example, offering access to a database of specific interest, such as a drug database for physicians at the Health Department). Our method allows for a discussion of both types. Domain-specific objectives require of course some customization of the method, often with the involvement of focus groups and site managers.

Site Structure. This represents the logical organization of the information and access structures. A well designed site has a structure reflecting a clear idea of its goals and the criteria according to which the published information is classified, and is accessible via its navigation paths. Therefore, a good structure facilitates access to the site contents, as the user can actually perceive the organization of information; while a loose and unclear structure disorients and confuses users.

In our method, in order to analyze the site structure, we hinge on the notion of *scheme*: the idea is to derive a high level description of the site which emphasizes the *page classes* and the network of navigation paths offered by the site. We use a simple and intuitive model (a simplified version of the ADM model to which we contributed; see Atzeni et al. [3]). We decided to adopt ADM because it allows the description of the site structure in terms of page classes, i.e. sets of pages sharing the same structure, and links, without detailing the structure of each page. As will be clarified in the following, this is a very important feature for our methodology. In addition, the adoption of a specific formal model for Web structure allows the application of Web *content* and *structure* mining techniques [3,5].

The quality of the site structure can be expressed by the evaluation of two features: the *structuring degree* and the *navigational organization*. In a Web site with a strong and clear structure, the number of pages similar in content and structure is much greater than the number of unique pages. The structuring degree is the ratio between the number of pages and the number of page classes. Clearly, this is not an absolute quantitative measure, because it should be considered against the specific domain and according to the objectives of the site, but the resulting figure helps the site managers to consider the restructuring of the site. In fact, in a site which provides access to thousands of pages: if the ratio is close to one, the published information tend to be organized in a very heterogeneous way; if conversely the number is very high, we have a highly structured site, where information is organized into a few very generalized page types.

The navigational organization concerns the structure of the data access paths. Since users mainly access a site by following hypertext links, it is important that the navigation paths are not too long: as some say, important pages should be only a few “clicks” away. The site scheme is also an effective means for evaluating this aspect, as it allows the verification of the main paths leading to each page. In the black-box approach, the site manager establishes a maximum length for each type of information, and checks if the actual paths satisfy this requirement.

More detailed analysis could be performed by relaxing the black-box approach in two ways: by involving focus groups, and asking them if the length of the paths is bearable, and by performing click stream analysis on the log files of the HTTP server and producing indicators of the relationship between path length and page access.

Site Services. In order to describe the information and the services offered by the site, it is often possible to refer to standards and classifications provided for specific sectors. For example, to describe site services in the public sector, we refer to the framework proposed by the European Union in the “Green Paper on Public Sector Information in the Information Society” [4], which classifies electronic services in three main categories:

1. *Information services* deal with the delivery of sorted and classified information;
2. *Communication services* allow the Public Administrations to interact with individuals (private or corporate) or groups of people;
3. *Transactional services* provide the means to acquire products or services on line or to submit data (e.g. government forms, voting).

The page classes of the site scheme are classified according to the above framework; thus, we obtain a grid that summarizes the site services. In particular, the first and third categories directly deal with data delivering services. It is also worth noting that these categories are general enough to be applied to any other sector without loss of completeness.

Site Effectiveness. Is aimed at verifying that the site objectives are satisfied. In our method, the results of the above points (objectives, structure, offer) are compared to evaluate the overall coherence and to find inconsistencies and weaknesses. The declared objectives are classified as for the analysis of the site services, and a check is performed between them and the actual services. In addition, for each offered service, we check if the corresponding pages are reachable within the established maximum number of links.

2.2 Specific Analysis

While the general analysis provides an overall picture of the site, the specific analysis concentrates on features that characterize individual aspects of a site. The results of the specific analysis should emphasize, in a given site, which features need to be improved to enhance the overall quality of the site. Also, they can be used in order to compare the quality of different sites.

In our method, we have identified several elementary attributes (about 50) to describe the basic features that contribute to the quality of a public sector Web site. Many of these attributes refer to features that play an important role for the quality of any Web site; others have been suitably introduced to catch aspects that assume a particular importance in the specific context of the public sector.

The method thoroughly details the criteria according to which each attribute must be measured. In fact, some attributes can be measured in an objective way, whereas others are subjective and also have different properties from page to page

(and therefore a reduced sample was needed, for obvious reasons). Therefore, for each attribute we have indicated:

- the sample on which it should be evaluated (a specific page, if applicable: for example, the quality of the home-page is evaluated on the home-page itself; in other cases, a random set of pages: for example to evaluate the simplicity of the user interface)
- the unit of measure (in some cases, something physically meaningful, for example Kbyte; in other cases a heuristic scale, such as 1-4)
- the evaluation method, which can be objective or subjective; in the latter case the method requires a set of analysts, each of whom gives a grade, and the assessment is obtained by calculating the average.

Features are grouped into three classes that refer to specific characteristics of a site: *Connectivity*, *Quality of the HTML Source Code* and *User-Site Interaction*.

Connectivity. It aims at estimating the efficiency of the site in managing the connections with the final user. This characteristic depends on the hardware, software and networking architecture, and on the size of the pages. Since the actual bottlenecks cannot be detected in a black-box analysis, our method concentrates on page sizes and in particular it focuses on the usage of images in the Web pages of the site. Besides estimating to what extent images contribute to the size of pages, the method requires to check the quality of their encoding (the use of images with a high dynamic should be avoided, since many browsers do not support large palettes). Also, since many Web browsers can show the text of a page yet while downloading its images only if the width and height attributes of each image are specified, the method controls that images are properly tagged into the HTML source code.

Quality of the HTML Source Code. It deals with the correctness of hyperlinks and HTML source code. The HTML code must be not only syntactically correct, but even semantically; there are tags (and attributes), such as the one that associates a title with the page, which should be used always and properly.

User-Site Interaction. The quality of the interaction of the user with the site can be defined under several different perspectives. Our method evaluates this crucial characteristic by concentrating on five different aspects, each associated with a set of suitable features. 1) *Characterization* deals with the features that allow the site to be easily recognized and remembered by the user; 2) *Functionality* is concerned with features that evaluate the quality and the quantity of provided functions; 3) *Quality of Content* includes features describing the offer of information and services of the site; 4) *Usability* is described by features that aims at catching the effectiveness and the efficiency of the interaction paradigms between site and users; finally, 5) *Accessibility* deals with features that give the site the ability to be browsed by any device, without discriminating any category of users (in particular disabled people as argued in the documents of the Web Accessibility Initiative [1]).

Attribute Name	Code	Sample	Unit	Evaluation
Connectivity				
Size of the Home Page (only HTML)	1	HP	kb	
Weight of images of the Home Page	2	HP	%	
Average size of the site's pages (only HTML)	3	LS	kb	
Average weight of images in the site's pages	4	LS	%	
Image Encoding	5	HP	heuristic	
Image Tagging	6	HP	heuristic	
Quality of the HTML Source Code				
Correctness of Hyper-Links	7	I	heuristic	
HTML code quality	8	LS	heuristic	
Page title quality	9	LS	heuristic	subjective
Site-User Interaction				
Characterization				
Identifiability	10	LS	heuristic	subjective
Graphical Characterization	11	LS	heuristic	subjective
Content quality				
Content coherence	12	LS	heuristic	subjective
Description of the administration and of its competences	13	I	yes/no	
Comprehensibility of the information	14	LS	heuristic	subjective
Hyper-contextualization of the information	15	LS	heuristic	subjective
Freshness of the Home Page	16	HP	heuristic	subjective
Historical identification of the information	17	LS	heuristic	
Declaration of the sources	18	LS	heuristic	
Authoritativeness of the sources	19	LS	heuristic	
Quality of the language	20	LS	heuristic	
Contents of the Home Page	21	HP	heuristic	subjective
Functionalities				
Presence of an internal search engine	22	I	yes/no	
Direct access to the internal search engine from the Home Page	23	HP	yes/no	
Functions for downloading hypertextual documents	24	LS	yes/no	
Printability of the pages	25	LS	yes/no	
Interaction with the site managers	26	I	yes/no	
Interaction with the Administration	27	I	yes/no	
Functions to detect customer satisfaction and collect comments	28	I	yes/no	
Security and Privacy	29	I	yes/no	
Usability				
Clarity of the user interface	30	LS	heuristic	subjective
Simplicity the user interface	31	LS	heuristic	subjective
Degree of standardization of user interface	32	LS	heuristic	subjective
Effectiveness of graphic layout	33	LS	heuristic	subjective
Presence of navigation bars	34	LS	yes/no	
Presence of tools to support navigation (TOC, maps)	35	I	yes/no	
Quality of tools supporting navigation	36	specific	heuristic	subjective
Instructions, tutorials, helps about the site usage	37	I	yes/no	
Functions for management of errors	38	I	yes/no	
Presence of versions in different languages	39	I	yes/no	
Accessibility				
Browser independence	40	LS	heuristic	
Applet/script-free access	41	LS	heuristic	
Image independence	42	LS	heuristic	
Textual navigator access	43	LS	heuristic	
Access from low resolution monitors	44	LS	heuristic	
Visibility on the major search engines	45	LS	heuristic	
Accessibility for people with disabilities	46	LS	heuristic	

LS=Large Sample (about 20 pages)

I= Site Inspection

HP= Home Page

Subjective Evaluations are given by the average of different analysts results

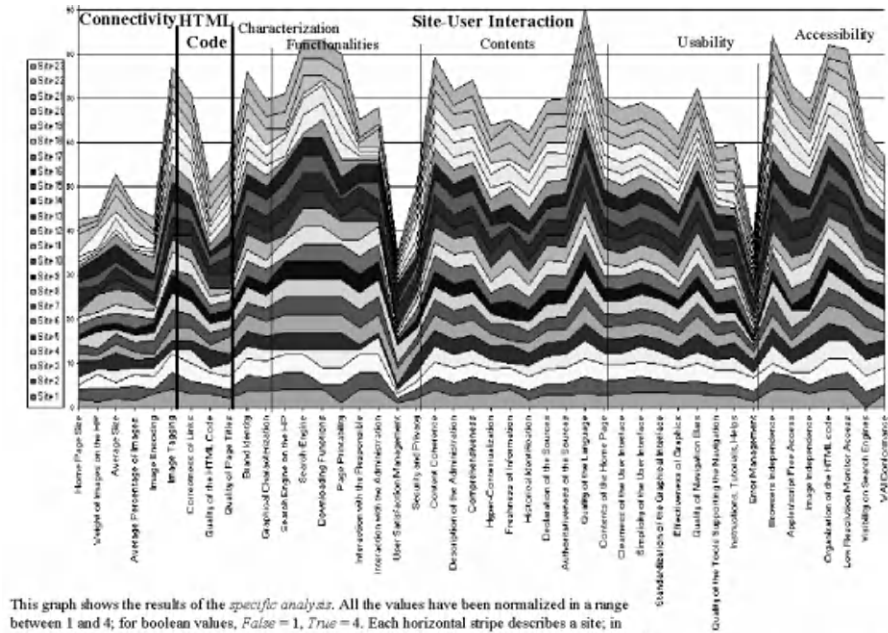
Fig. 1. The attributes of the Evaluation Method

Figure 1 summarizes our method; it describes the various attributes we take into account and sketches the associated measurement criteria (an exhaustive description is out of the scopes of the paper): the unit of measure, the sample of pages to be involved in the measure, and the kind of evaluation (subjective/objective).

3 A Case Study: Evaluation of the Web Sites of the Italian Government

We have tested the method in the analysis of 23 sites of the Italian Public Administration. In this section we briefly report on this activity, by discussing the overall results of the general analysis, and taking some relevant elements from the specific analysis, which is summarized in the graph of Figure 2. Note that the uniform scoring scale of the attributes does not mean that we want to actually compare their values. The aim of the diagram is to give a pictorial impression of the overall state of the art of the analyzed sites with respect to the attributes.

This case study served also as a first validation benchmark for the selected criteria, the majority of which proved to be easy to assess. One major remark is worth making for those attributes, like for example the “Quality of the language”, which, in some specific subject matters, would require the involvement of specific skills to be appropriately evaluated.



This graph shows the results of the *specific analysis*. All the values have been normalized in a range between 1 and 4; for boolean values, *False* = 1, *True* = 4. Each horizontal stripe describes a site; in every point, the width of a stripe depends on the value of the corresponding feature shown on the x-axis; therefore, the wider is the stripe associated with a site, the better is the quality of that site. Also, the graph distinguishes regions corresponding to the main characteristics of our evaluation method. The more a region is filled, the higher is the quality of the corresponding quality dimension.

Fig. 2. Results of the Specific Analysis on the Sites of the Italian Public Administration

The use of the ADM logical model to represent the structure of the analyzed sites allowed us to put into evidence some aspects which should be improved. For example, we observed that the structuring degree of the sites is satisfactory,

whereas the navigation paths of several sites are tortuous, and many pages are too many “clicks” away from the home page. We observed that this phenomenon is mainly due to the fact that the structure of these sites reflects the organization of their administration in some way. Most of these sites exhibit a weak content coherence and a confusing navigation interface, with a consequent worsening of the site usability and data accessibility.

The analysis of the contents gave a satisfactory overall quality; as can be seen in the corresponding area of Figure 2. The information content is coherent and the quality of the language is high (corresponding to the maximum on the graph). Nevertheless, the details of the specific analysis also reveal that, among the features depicting the site content quality, the historical identification of the delivered information, and the declaration and authoritativeness of the information sources all have low scores.

The communication services provided by the sites of our study are normally limited to the possibility of sending e-mail messages to the administration. However, with this respect, a negative note, which comes from the specific analysis, indicates that only a few of the investigated sites provide specific functions for collecting the users satisfaction and statistics about the quality and the effectiveness of this service.

We encountered a few experiences of transactional services, mostly limited to specific initiatives, or dedicated to small communities of users. However, it is important to remember that a key issue for the development of on-line government services is legislation, and important bills (such as the law on electronic signature and the appointment of certification agencies) have been only approved very recently in Italy.

Web sites should not inhibit the access by any category of users. The results of the specific analysis dealing with accessibility show that access to a large fraction of the sites is independent from the browser (the local maximum peak corresponds to this feature). Nevertheless, although the organization of the HTML source code is satisfactory even for textual browsers, the use of images, applets, and scripts heavily limits the accessibility of several sites.

Other details from the specific analysis indicate that connectivity is a weak point for most of the sites. In particular it is worth observing that images significantly augment the size of the pages, and that their encoding is rarely optimized for the Web. Moreover, the analysis of features related to the HTML code shows that the presence of missing links does not affect the sites of our study; however, the HTML source code contains several errors, and the quality of the page titles is low (often they are not used at all).

These results show that, by using the proposed evaluation methodology, it is possible to assess and compare the overall quality of a large number of Web sites and to emphasize the aspects which are more closely related with the quality of the provided information. A group of around 10 analysts was able to complete the assessment of the 23 sites in about 40 working days.

The overall conclusion is that the sites we have analyzed certainly satisfy their minimal objectives. Nevertheless, our impression is that the development of several sites still follows an amateurish approach, which is not acceptable if compared with the challenges posed by the e-government plans of many coun-

tries. The life-cycle of these sites still follows primitive, not very well planned patterns. In particular, the editorial boards, when they exist, are not adequately structured in terms of both resource quantity and skills. This scenario, when the sites will grow bigger, may lead to a fast drop of their quality.

4 Discussion

A first major finding of our study was the need for a specific approach to Web site quality for the public sector (or, at least, an approach that is not based only on profit nor on number of accesses; and obviously not based on even more superficial issues such as graphics quality). Quality here has to take into account different aspects, and each of them should be pursued in parallel. This probably also applies to complex business sites, with each of the features being instrumental in pursuing the “profit” approach. However, in the public sector, for ethical, or political, or legal issues, the requirements are more stringent. Also, decisions often require more time than for private businesses and therefore the problem becomes even more complex.

The only way to pursue such an ambitious objective is to follow a disciplined approach that includes quality since the beginning as a major goal in each of the steps of the development of Web sites. This should start as early as in the strategic planning phase. Indeed, since the Web is a primary means of communication (and will soon become fundamental in delivering services) it is clear that a top-level commitment is an essential condition for achieving success. Also, integration with other forms of communication, with possible re-engineering is needed. Strategic planning for Web sites becomes therefore crucial, in order to define goals, priorities, phases, and to prepare the way for evaluating results. Then, quality assurance should appear in each of the phases of the life-cycle: when designing the organization of a site very important quality aspects arise (most of those involved in the general analysis of our method); in the construction phase most specific aspects are taken into consideration; during the operational phase periodic evaluations should be performed, without ever assuming that quality has been achieved: the high dynamicity of the Web will always induce continuous changes in requirements and usage modes. In this respect various types of assessments can be performed, and our approach allows for both black-box ones, fast and cheap, as well as for more detailed ones, with personalization and involvement of user groups.

To the best of our knowledge, there is a small number of papers that explicitly deal with the quality of Web sites [7,11]). Gregg extends models proposed in the framework of databases to the context of the Web [7]). Compared to this, we certainly lack some specific aspects (for example, we do not consider access security). Olsina *et al.* propose a Web site evaluation method that is similar to our specific analysis; in addition, they define a method to gather a global evaluation rate.

References

1. Web accessibility initiative (WAI). <http://www.w3.org/WAI>.
2. M. L. Abate, K. V. Diegert, and H. W. Allen. A hierarchical approach to improving data quality. *Data Quality*, 4(1):September, April 1998.
3. P. Atzeni, G. Mecca, and P. Merialdo. To Weave the Web. In *International Conf. on Very Large Data Bases (VLDB'97)*, Athens, Greece, August 26-29, pages 206–215, 1997.
4. European Commission. Public sector information: A key resource for europe — green paper on public sector information in the information society.
5. V. Crescenzi, Mecca G., and P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *International Conf. on Very Large Data Bases (VLDB'01)*, Roma, Italy, September 11-14, pages 109–118, 2001.
6. M. Cutler and J. Sterne. E-metrics. business metrics for the new economy. Net-Genesis white paper., 2000. <http://www.netgen.com/emetrics/>.
7. D. G. Gregg. Supporting web content quality: Formalizing meta-data concepts for the web-domain. In *ICIS 1999*, 1999.
8. P. J. Lynch and S. Horton. *Web Style Guide: Basic Design Principles for Creating Web Sites*. Yale University Press, 1999.
9. J. Nielsen. Impact of data quality on the web user experience., July 1998. <http://www.useit.com/alertbox/980712.html>.
10. J. Nielsen. *Web Usability: the Practice of Simplicity*. New Riders, 1999.
11. L. Olsina, G. Lafuente, and G. Rossi. Specifying quality characteristics and attributes for websites. In *Web Engineering, Software Engineering and Web Application Development*, pages 266–278, 2001.
12. L. Rosenfeld and P. Morville. *Information Architecture for the World Wide Web*. O'Really, 1998.
13. B. Shneiderman. *Designing the User Interface*. Addison Wensley, 1998.
14. J. Spool, T. Scanlon, W. Schroeder, C. Snyder, and T. DeAngelo. *Web site usability: A designer's guide*. Morgan Kaufman, 1998.
15. R. Y. Wang, D. M. Strong, and L. M. Guarascio. Data consumers' perspectives of data quality. Total data quality management (TDQM) research program report, Massachusetts Institute of Technology, 1994.

Automatic Web Information Extraction in the ROADRUNNER System

Valter Crescenzi¹, Giansalvatore Mecca², and Paolo Merialdo¹

¹ D.I.A. – Università di Roma Tre

² D.I.F.A. – Università della Basilicata

Abstract. This paper presents ROADRUNNER, a research project that aims at developing solutions for automatically extracting data from large HTML data sources. The target of our research are *data-intensive* Web sites, i.e., HTML-based sites with a fairly complex structure, that publish large amounts of data. The paper describes the top-level software architecture of the ROADRUNNER *System*, and the novel research challenges posed by the attempt to automate the information extraction process.

1 Introduction

Extracting data from HTML pages and making them available to computer applications is becoming of utmost importance for developing several emerging e-services. Many new classes of applications aim at leveraging on the huge amount of data delivered on the Internet as HTML pages. For example, a service which is becoming quite popular on the net is that provided by the so called *shopping agents*, that is software modules that navigate the network looking for better prices of specific items on e-commerce sites. Another important example is represented by the needs of modern portals, whose goal is offering integrated access to several Web services (like home banking, on-line trading, billing, credit card checking etc.) from a single interface.

These applications need to manipulate data delivered into HTML pages by several Web sites. Nevertheless, since the target consumers of Web data sources are humans – rather than machines – the data encoded into HTML pages cannot be directly handled by applications: they need to be extracted from HTML pages and made comprehensible to machines. Data extraction from HTML Web pages is usually performed by software modules called *wrappers*, i.e. programs that are able to extract data items from HTML pages and return them in a structured format (e.g. in XML). However, writing wrappers for HTML by hand may in some cases be a labor-intensive task. Also, manually coded wrappers are usually quite brittle, since even small changes at the site may prevent the wrapper from working properly. Therefore, maintaining applications that need to process data extracted from the Web becomes a costly and complex issue.

ROADRUNNER is a research project that aims at developing solutions for automatically extracting data from large HTML data sources. This paper presents an overview of the project and describes the top-level software architecture of

the ROADRUNNER *System*, which has been specifically designed to automate the data extraction process.

The paper is organized as follows. First, Section 2 gives an overview of the project goals, trying to convey an intuition of the key ideas and of the original features with respect to other proposals in the literature. Section 4 describes the overall architecture of the ROADRUNNER system. The following sections then concentrate on the description of the various modules.

2 Overview

The target of our research are *data-intensive* Web sites, i.e., HTML-based sites that publish large amounts of data in a fairly complex structure. Data are usually stored in a back-end DBMS, and HTML pages are dynamically generated using *scripts* from the content of the database; simply speaking, these scripts run queries on the database – possibly nesting the original relational tables – and return the result-set in HTML format. Roughly speaking, we may say that each script generates a *class of pages* with homogenous content and layout. These sites usually contain different *classes of pages*, corresponding to different contents in the site, like, for example, pages about MP3 players sold on *amazon.com* or pages about stock quotes on *yahoo.com*. Our goal is that of running database-like queries on these pages, like “name and brand of the MP3 player with the lowest price” or “last variation of Microsoft stock certificates”.

To do this, we follow a two-step approach: (i) we provide a wrapper layer that extracts relevant pieces of information from the original HTML pages in the sites, and returns them as a collection of XML documents on the same DTD or XMLSchema; to give an example, the XML documents derived for pages about MP3 players on *amazon.com* might contain a list of `<player>` elements, each with `<name>`, `<brand>`, `<price>` subelements, plus an optional sublist of comments and ratings written by customers who purchased that player. (ii) Then, based on these wrappers, an XML query language (like XQuery [5], or any other of the other languages that have been recently proposed [3]) can be used to express and evaluate queries on these derived XML documents.

Given the availability of several XML query languages, it can be seen that most of the complexity of this process lies in the generation of wrappers that perform the translation from HTML to XML. In light of this, in this paper we concentrate primarily on this problem, and assume that an external XML-query language engine is available to run queries on the resulting XML docs.

2.1 Supervised Grammar Inference

Generating a wrapper for a set of HTML pages corresponds to deriving a grammar for the HTML code in the page – usually a regular grammar – and then use this grammar to parse the page and extract pieces of data. Grammar inference is a well known and extensively studied problem (for a survey of the literature see, for example, [18]). However, regular grammar inference is a hard problem: first,

it is known from Gold's works [10] that regular grammars cannot be correctly identified from positive examples alone; also, even in presence of both positive and negative examples, there exists no efficient learning algorithm for identifying the minimum state DFA that is consistent with an arbitrary set of examples [11]. As a consequence, the large body of research that originated from Gold's seminal works has concentrated primarily on finding efficient algorithms for *supervised* grammar learning, i.e., algorithms that work in presence of additional information, typically a set of labeled examples or a knowledgeable teacher's responses to queries posed by the learner.

The fact that regular expressions cannot be learned from positive examples alone, and the high complexity of the learning even in presence of additional information have limited the applicability of the traditional grammar inference techniques to Web sites, and have recently motivated a number of pragmatic approaches to wrapper generation for HTML pages. These works have attacked the wrapper generation problem under various perspectives, going from machine-learning [9,20,15,13,16] to data mining [1] and conceptual modeling [8, 19]. Although these proposals differ in the formalisms used for wrapper specification, they all inherit the "supervised learning" approach of traditional grammar inference, and share a number of common features:

- *manual selection of sample pages*: first, it is assumed that a human supervisor selects a collection of homogeneous HTML pages from which data should be extracted; the wrapper generation system usually derives the wrapper by working on a single page at a time; then, the wrapper is used to parse the remaining pages, possibly generalizing the grammar when this is needed;
- *availability of labeled examples*: second, the wrapper generator works by using additional information on the content of the page under exam, typically a set of labeled samples provided by the user or by some other external tool; the wrapper is inferred by looking at these positive examples and trying to generalize them;
- *a priori knowledge about the target schema*: third, it is usually assumed that the wrapper induction system has some *a priori* knowledge about the page organization, i.e., about the schema of data in the page; most works assume that the target pages contain a collection of flat records; in other cases [1] the system may also handle nested data, but it needs that the user specifies what are the attributes to extract and how they are nested.

In essence these works deal with a data extraction problem that we may summarize as follows: "given one or more HTML pages, a target schema (usually a list of flat records) for these pages, and a set of labeled examples, find a set of extraction rules that allow to parse the HTML code and retrieve data items according to the target schema".

2.2 Our Approach

Our research departs quite significantly from these proposals and investigates the wrapper generation problem under a new perspective. In particular, we aim

at automating the wrapper generation process to a larger extent; for this reason, we investigate *unsupervised* wrapper generation, as follows:

- first, *we do not assume that sample pages are manually selected* from a human designer; on the contrary, we assume that the system is able to automatically cluster pages in a (portion of a) site into homogeneous classes;
- second, *our system does not rely on user-specified labeled examples*, and does not require any interaction with the user during the wrapper generation process; this means that wrappers are generated and data are extracted in a completely automatic way;
- finally, *we do not assume any a priori knowledge about the target schema*, i.e., our system does not know the schema according to which data are organized in the HTML pages: this schema will be inferred along with the wrapper; moreover, our system is not restricted to flat records, but can handle arbitrarily nested structures;

In essence, we may say that, with respect to other works in the literature, we deal with a different problem, a *schema finding and data extraction problem* [12], which we may summarize as follows: “given a set of HTML pages, find a schema (arbitrarily nested) for the content of these pages, and a set of extraction rules that allow to parse the HTML code and retrieve the data according to the discovered schema”. This change of perspective on the overall problem of information extraction from Web pages represents the main source of originality of this research.

In fact, on the one side, this forced us to reconsider the problem of grammar inference for HTML pages, and devise new techniques to deal with it; the main intuition behind our proposal is that pattern discovery can be based on the study of similarities and dissimilarities between the pages; in particular, we have proposed [6] a novel approach to wrapper inference for HTML pages, in which, in order to tell meaningful patterns from meaningless ones, our system works with two HTML pages at a time, and mismatches are used to identify relevant structures.

On the other side, we had to face a number of side research problems that had not been considered before, since the presence of user-provided inputs made them irrelevant. These problems go from that of selecting collections of homogeneous pages in a site, i.e., pages with the same structure, to that of labeling attributes in the target schema once this has been derived (i.e., stating that string “\$15.99” is a price, and “Java for Dummies” is a book title).

In the following sections we describe the various modules that compose the architecture of our system, and how these concur to the solution of these problems.

3 The Wrapper Generation Algorithm

In our approach a wrapper is a regular grammar that can be parsed against an HTML page to retrieve some data items. To be more precise, given an alphabet

of terminal symbols Σ , and a set of non-terminal symbols N , we consider a subset of the regular grammars, corresponding to regular expressions built over $\Sigma \cup N$ using the following operators (ϵ is the empty sequence): (i) concatenation, of the form $a \cdot b$, i.e., the sequence of a and b ; (ii) iteration, of the form a^+ , i.e., the repetition of a one or more times; (iii) “hooks” of the form $(a)?$, a shortcut for $a|\epsilon$.

For example, the following regular expression may define a wrapper in our formalism ($\$$ denote non-terminals):

```
<HTML>...
  Player: <B>$Name</B> $Brand - our price: <STRONG>$Price</STRONG>
         <UL> (<LI> <FONT>$Rating</FONT> (<I>$Review</I>)?)+ </UL>
...</HTML>
```

Note that, with respect to general regular grammars we allow for a very limited form of disjunction, basically only the ones hidden inside hooks (zero *or* one), and inside iterations (one *or* more). In essence, our wrapper corresponds to union-free regular grammars. This, of course, introduces a limitation in the expressive power of the wrapping language. However, our experience tells that the language includes many of the typical patterns that occur in fairly structured HTML sources [6].

Since wrappers essentially parse the HTML code, they are targeted at a specific page organization and layout. Therefore, in order to extract data from a site, we need to derive one wrapper for each page class in the site. In ROAD-RUNNER, this decoding activity is based on the similarities that are exhibited by HTML pages belonging to the same class, i.e., we try to infer a grammar for a class of pages by looking at a number of samples, and then use this grammar as a wrapper.

We have recently developed an original technique to infer a wrapper for a class of pages by analyzing similarities and differences among some sample HTML pages of the class [6]. In essence, given a set of sample HTML pages, our technique compares the source HTML codes, in order to find matching and mismatching parts and, based on this knowledge, progressively refines a common wrapper. The output wrapper is a grammar that can be parsed against the pages of the class to extract data items. These ideas are clarified in Figure 1, which refers to a fictional bookstore site. In that example, pages listing all books by one author are generated by a script; the script first queries a database to produce a nested dataset (Figure 1.a) by nesting books and their editions inside authors; then, it serializes the resulting tuples into HTML pages (Figure 1.b). When applied on these pages, the Aligner technique compares the HTML codes of the two pages, infers a common structure and a wrapper, and uses that to extract the source dataset. Figure 1.c shows the actual output of the current implementation of the Aligner after it is run on the two HTML pages in the example. (For the sake of readability, the extracted dataset is produced in HTML format. As an alternative, it could be formatted in XML, or stored in a database.)

Several things are worth noting here. First, since the matching technique is based on the comparison of pages of the same type, one critical assumption is that HTML pages coming from the site have been somehow clustered into the

a. Source Dataset

Name	Books				
	Title	Descr.	Editions		
			Details	Year	Price
John Smith	Database Primer	This book...	First Edition, Paperback	1998	20\$
			Second Edition, Hard Cover	2000	30\$
Paul Jones	Computer Systems	An undergraduate...	First Edition, Paperback	1995	40\$
	XML at Work	A comprehensive...	First Edition, Paperback	1999	30\$
	HTML and Scripts	A useful HTML...	null	1993	30\$
	JavaScript	A must in...	Second Edition, Hard Cover	1999	45\$
			null	2000	50\$
...	...	...	...	...	...

b. HTML Pages

www.csbooks.com/author?John+Smith



www.csbooks.com/author?Paul+Jones



c. Data Extraction Output

Total number of SCHEMAs found: 1					
Schema Number 1: A (B ((C) ? D E) * F) * Total Time: 0* 180 ms					
sample1.html					
A	B	C	D	E	F
John Smith	Database Primer	First Edition, Paperback	1998	\$20	This book introduces the reader to the theory and technology...[TRUNCATED]
		Second Edition, Hard Cover	2000	\$30	
	Computer Systems	First Edition, Paperback	1995	\$40	An undergraduate level introduction to computer...[TRUNCATED]
sample2.html					
A	B	C	D	E	F
Paul Jones	XML at Work	First Edition, Paperback	1999	\$30	A comprehensive description of XML and all related standards ...[TRUNCATED]
	HTML and Scripts	null	1993	\$30	A useful HTML handbook, with a good tutorial on the use of so...[TRUNCATED]
	JavaScripts	null	2000	\$50	A must in every Webmaster's bookshelf ...

Fig. 1. Examples of HTML Code Generation

different classes they belong to; then, to support the above algorithms, there is the need to adopt some clustering techniques for HTML pages that allow to quickly classify pages based on their type; these techniques should aim at giving

a good approximation of the page classes in the site, in order to carry on the decoding step.

Second, sites usually also contain singleton pages, i.e., pages such that there is no other page in the site with the same organization and layout; the Home Page of a site usually falls in this category. These pages mainly serve the purpose of offering browsable access paths to data in the site; the *Aligner*, which is based on comparing two or more pages and finding similarities, obviously is not applicable to these singleton pages; therefore there is the need to develop specific techniques for wrapping these pages also;

Finally, as it can be seen from the example in Figure 1.c, whenever we infer a nested schema from the pages belonging to one class, the inferred schema has anonymous fields (labeled by A, B, C, D, etc. in our example). In order to enhance the semantics of the schema, each field should be associated with a meaningful name. This step could be done manually after the dataset has been extracted. Nevertheless, since our ultimate goal is that automatizing the whole data extraction process, one intriguing alternative is to develop some form of post-processing of the wrapper to automatically discover attribute names.

The software architecture of the ROADRUNNER system, discussed in the next section, is aimed at providing solutions to all of these problems; it is centered around the *Aligner*, i.e., the module that implements the wrapper generation algorithm, and complements it with a number of additional modules.

4 The ROADRUNNER System Architecture

Figure 2 shows the top-level architecture of the ROADRUNNER system.

We identify four main modules, each of which addresses a specific problem, as follows:

- the *classifier* analyzes pages from the target site and collects them into clusters with a homogeneous structure, i.e. it tries to identify the page classes offered by the site. this module incorporates a crawler that navigates the target site, and exploits a number of heuristics for producing a good approximation of the page classes in the site; note that some of these classes may contain several candidate pages and will be fed to the *Aligner* for wrapper generation; other classes will have singleton elements;
- wrapper generation for classes of similar pages is performed by the *Aligner*, which implements the matching technique; for each class of pages, the *Aligner* compares the HTML sources of some sample pages to infer a grammar to be used as a wrapper for the whole class;
- classes having singleton pages are fed to a module called *Expander*, which tries to infer a wrapper for them; wrappers generated by this *Expander* are based on different techniques with respect to those inferred by the *Aligner*;
- finally, the *Labeler* associates a semantic meaning to the data fields that can be extracted by running the wrappers generated by the above modules against the site pages; i.e. it gives an appropriate name to each non-terminal symbol of the wrapper grammar.

In the following we briefly illustrate the main features of the various components.

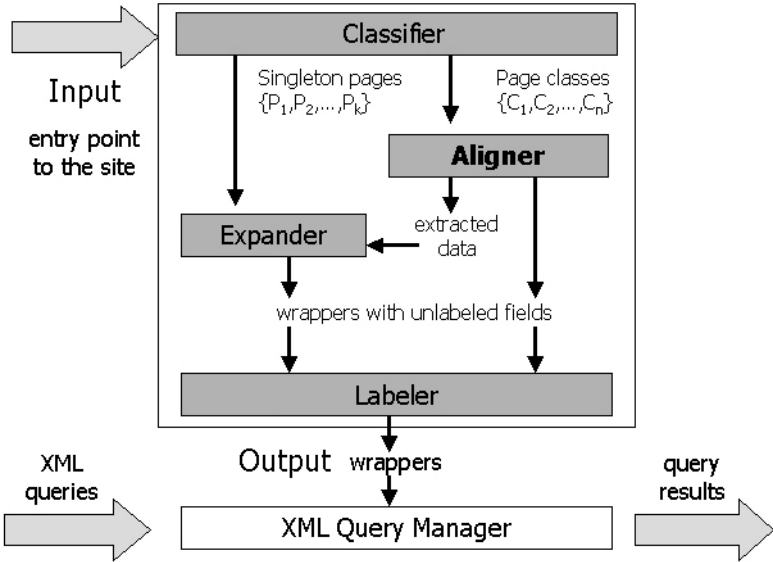


Fig. 2. Architecture of the System

4.1 The Aligner

The Aligner implements the ACME technique, which represents the core of our system. In the following we sketch a description of the ACME technique. For a deep description we refer the interested reader to [6], which also reports the results of several experiments.

ACME takes as input HTML page sources as list of tokens (a lexical analyzer transforms pages into lists of tokens); each token is either an HTML tag or a string value. Then ACME works on two objects at a time: (i) a list of tokens, called the *sample*, and (ii) a wrapper, i.e., one union-free regular expression. Given two HTML pages, to start we take one of the two, for example the first page, as an initial version of the wrapper; then, the wrapper is progressively refined trying to find a common regular expression for the two pages. This is done by progressively solving *mismatches* between the wrapper and the sample. The wrapper can then be further refined by iteratively applying the same technique over the samples of a collection of HTML pages of the same class. Our experiments show that a small number of sample pages (3-6) is sufficient to infer a grammar wrapper for all the pages of a large class.

5 The Classifier

The goal of the Classifier [7] is to efficiently identify the different pages classes in the target sites. For each class, a number of samples will be given as input to the Aligner in order to generate the corresponding wrappers. The Classifier crawls a site in order to cluster its HTML web pages according to the grammar they obey to. The clustering process is based on known techniques. Roughly speaking we may say that these techniques see the samples to cluster as points in a n -dimensional real space, usually called the *feature space*; cluster are then generated trying to minimize the average distance between points in a cluster (see [14] for a survey on the topic).

One key point in this process is choosing the right mapping of a sample to the feature space. This is done by extracting some relevant (numeric) *features* from the sample and using them as coordinates in the feature space. In our context, determining useful features becomes an intriguing problem, completely different from the classical ones. In fact, we have a radically new notion of similarity between our samples, that is, the compliance to a common regular grammar. To give an intuition of how this new notion of similarity makes our clustering problem different from the classical string clustering problem, consider the following: (i) usually two strings are considered similar if they contain common sub-parts. On the contrary, in the Web page case, requiring that two pages contain common sub-parts is not sufficient to guarantee that they can be parsed using the same grammar – this requires a deeper form of structural similarity; (ii) also, typically strings that are considered similar by classical methods have approximately the same length; in our case, pages in the same class may have largely different sizes due to different cardinalities of patterns under a Kleene’s star; consider for example two pages about books, each with the list of books by a certain author; if the first author has published few books (say 3) and the second one many more (say, 15), the sizes of the two HTML sources may be completely different; nevertheless, the two pages are very likely to belong to the same class.

We have identified two families of features that can give information about the similarity of pages as needed in our context. In the former family, there are features that aims at giving information about the internal structure of the pages; clearly these properties are somehow connected to the structure of the common grammar these pages should obey to. In the current implementation we have taken into account the following features:

- *Tag Probability* it is reasonable to assume that pages complying the same grammar have a similar “distribution” of tags, i.e., tags appear in the pages with similar probability; we have therefore considered tag probabilities as possible features for the Web page clustering problem; in practice, we calculate for each sample the percentage of occurrences of each tag with respect to the total number of tags, and use these numbers as coordinates in the feature space.
- *Tag Periodicity* there are cases in which tag probabilities may be misleading, since they do not give information about the relative positions of tags. This means that two pages containing approximately the same tags will be

considered similar even if the tags are completely rearranged with respect to each other, and therefore the pages are different in terms of the grammar. To complement tag probability with some other feature that gives us information on tag positions, we apply to HTML pages a variant of the classical frequency spectrum method [17].

- *Distance from the Home Page* if navigation paths in the site are well organized, it is reasonable to assume that pages containing homogeneous information are approximately at the same distance from the home page in the site graph.
- *URL Similarity* experience tells that, in most sites, URLs of pages in the same class follow some common patterns, either due to the fact that the HTML files are stored in a common physical folder of the server, or that the pages are generated by the same script. To take this into account, we use the classical notion of string similarity [2] in order to associate a number of numerical features with each URL string.

Interestingly, our experiments show that the four classes of features together, usually guarantee a good approximation of the correct classification.

6 The Expander

The Expander is responsible for extracting data from singleton pages, i.e. pages forming a separate class by themselves, so that each class is composed of a unique instance. It is worth noting that usually the role of these pages in a site is to collect access paths, i.e. links, to other pages; usually, the anchor of these links is a value that is repeated in the destination page. For example, consider a bookstore site; a possible singleton page in the site is the one that presents a list of all the book genres sold by the store. This page offers links that lead to pages presenting books of a given genres. For the sake of usability, the names of the various genres compare both in the singleton page and in the destination pages.

On the basis of these observations, we do not need extract data from singleton pages; they works as indices in a database, and the data items they contain are redundant with those provided by other (richer) pages. Therefore, they do not carry any useful information.

However, these pages might be useful to efficiently maintain the extracted data up-to-date with the Web source. So far we have described the data extraction as a one-shot process made by the following steps: 1) download all the site pages, 2) build wrappers for them, 3) apply the wrapper to extract relevant data. In order to keep data up-to-date, we should periodically repeat this process, every time downloading the whole site.

An alternative strategy is to leave data in the site, and to build a *virtual dataset*. Whenever one query is issued against the dataset, the system has to navigate the site and extracts data relevant to the query. Singleton pages providing access paths can efficiently support this approach. Therefore inferring a wrapper also for singleton pages may become a relevant issue. It is worth noting that the problem here has specific objectives: we are not interested in extracting

data provided by these pages; our objective is to build a wrapper for these pages in order to allow the system to navigate the whole site.

This goal can be achieved by applying wrapper inference techniques developed in other contexts. The crucial point is that we have at disposal the data offered in singleton pages: we have seen that they offer redundant data, that is data which can be extracted elsewhere in the site (applying the matching technique). In other words, we have positive examples to infer the wrapper grammar. This is the task addressed by several techniques known in the literature, such as, for example, Nodose [1], or Stalker [16]. Thus, we have planned to integrate one of these techniques into the ROADRUNNER architecture in order to support the Expander.

7 The Labeler

The Aligner is able to infer a wrapper and its associated schema for a class of pages; however, as we have said above, non-terminal symbols, which corresponds to attributes over the schema, are assigned anonymous names, as shown in the example of Figure 1.c.

The goal of the Labeler is to associate a meaningful name to each attribute of the extracted data set. Clearly this step could be done manually; however, in order to automatize every facet of the data extraction process we are currently investigating techniques to discover an appropriate name for each field.

One possible solution to the problem relies on the adoption of knowledge representation techniques: analyzing the extracted data, and exploiting the knowledge managed by some domain ontology, it may be possible to deduct some meaning for the fields. However, it is worth observing that important information about data is available on the Web pages themselves. Since Web sites are intended to be browsed by humans, it is a common practice that the data published into HTML pages are accompanied by textual descriptions to help the user directly and correctly interpret the underlying information. In many cases these descriptions are indispensable to correctly present data to the user. For example, consider how prices are presented in e-commerce Web sites: without the help of strings such as "our price" and "saving", pricing data would be completely misunderstood. Also, textual descriptions are often associated to data items organized into tables; usually, the first rows reports labels to describe the contents of the columns (the usage of table headers is strongly suggested by the W3C itself for the sake of usability).

Based on these observations we are setting up several methods to analyze the HTML code of the sample pages processed by the Aligner in order to find strings that represent good candidates for naming the fields of the data set. Our methods are based on a generalized notion of closeness between wrapper's tokens and non-terminal symbols. In particular, we deal with the twofold nature of an HTML wrapper: on the one hand the wrapper can be seen as a sequence of symbols; on the other hand, since the wrapper keeps the nested structure of an HTML document, it can be seen as a DOM tree as well, introducing special nodes to denote iterations and hook. Figure 3 illustrates this concept; note that the marked node represents the iteration.

```
... <TABLE> <TR> <TD> Name </TD>
<TD> Phone </TD> </TR>
(<TR> <TD> $A </TD> <TD> $B </TD> </TR> )+
</TABLE> ...
```

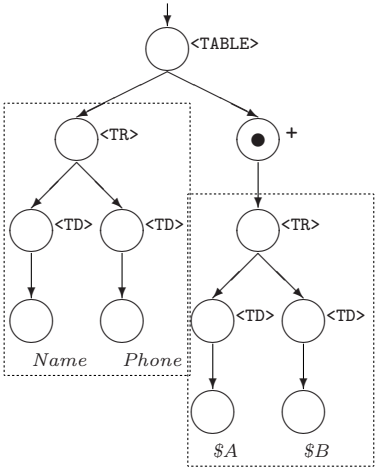


Fig. 3. Wrappers as DOM Trees

When searching for the name of a given field the Labeler explores the sequential representation of the wrapper, looking for a text string which is adjacent to the non-terminal. The "adjacency" is defined with respect to this specific context, and several characteristics of the HTML format concur at defining this properties.

When analyzing non-terminal symbols which are included in a repeated pattern, the Labeler also considers the DOM tree representation of the wrapper. The idea here is to check whether the pattern sub-tree is adjacent with some isomorphic sub-tree. In this case, the leaves of the discovered tree can be selected as names for the non-terminals of the pattern tree. Also in this case the notions of adjacency and isomorphism are defined with respect to our specific context.

Figure 3 gives an intuition of this strategy. The left dashed tree corresponds to a heading for the iterated pattern represented by the right dashed tree: since the two are isomorph the leaves of the former – namely, the strings "name" and "phone" – are candidate to be used as names for the non-terminals \$A and \$B respectively.

These methods assume that for each field a description is present in the page (and then into the wrapper grammar). However, there are many situations in which data are published in the Web page leaving their meaning implicit. For example, in a page presenting the details of a sold book, it might not be necessary to explicitly associate the book's title a string to describe that what follows is

the book title; it is assumed that the user is able to interpret the right semantics to that data item. Clearly, in these cases the above techniques fail.

To face these cases, we are currently studying an approach that relies on the richness of the Web itself, and which is inspired to Brin's DIPRE technique [4]. The Web can be considered as a huge knowledge base, where a particular piece of information may be published by thousands of independent sites. Clearly each site may adopt different presentation policies. Therefore, with respect to our specific problem, it is possible that in some page a given data item is associated with some information describing its meaning. Intuitively, for those non-terminal symbols whose names have not been derived by the above techniques, the system could extract some sample data item, and give them as input to a Web search engine. It is reasonable that in some of the pages retrieved by the search engine, the input value is explicitly associated with some descriptive text.

References

1. B. Adelberg. NoDoSE – a tool for semi-automatically extracting structured and semistructured data from text documents. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD'98)*, Seattle, Washington, 1998.
2. R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In *International Conference of Foundations of Data Organization (FODO'93)*, pages 69–84, 1993.
3. A. Bonifati and S. Ceri. Comparative analysis of five XML query languages. *ACM SIGMOD Record*, 29(1):68–79, 2000.
4. D. Brin. Extracting patterns and relations from the World Wide Web. In *Proceedings of the First Workshop on the Web and Databases (WebDB'98) (in conjunction with EDBT'98)*, pages 102–108, 1998.
5. D. Chamberlin et al. Xquery 1.0: An xml query language. W3C Working Draft, June 2001.
6. V. Crescenzi, G. Mecca, and P. Merialdo. ROADRUNNER: Towards automatic data extraction from large Web sites. In *International Conf. on Very Large Data Bases (VLDB'2001)*, Rome, Italy, September 11–14, pages 109–119, 2001.
7. V. Crescenzi, G. Mecca, and P. Merialdo. Wrapping-oriented classification of Web pages. Submitted for publication, 2001.
8. D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, Y. Ng, D. Quass, and R. D. Smith. A conceptual-modeling approach to extracting data from the web. In *Proceedings of the 17th International Conference on Conceptual Modeling (ER'98)*, pages 78–91, 1998.
9. T. Goan, N. Benson, and O. Etzioni. A grammar inference algorithm for the world wide web. In *AAAI Spring Symposium on Machine Learning in Information Access*, 1996.
10. E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
11. E. M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.
12. S. Grumbach and G. Mecca. In search of the lost schema. In *Seventh International Conference on Data Base Theory, (ICDT'99)*, Jerusalem (Israel), *Lecture Notes in Computer Science*, Springer-Verlag, pages 314–331, 1999.

13. C. Hsu and M. Dung. Generating finite-state transducers for semistructured data extraction from the web. *Information Systems*, 23(8):521–538, 1998.
14. A. K. Jain, N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
15. N. Kushmerick, D. S. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *International Joint Conference on Artificial Intelligence (IJCAI'97)*, 1997.
16. I. Muslea, S. Minton, and C. A. Knoblock. A hierarchical approach to wrapper induction. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 190–197, 1999.
17. A. V. Oppenheim, R. W. Schaffer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, second edition edition, 1999.
18. L. Pitt. Inductive inference, DFAs and computational complexity. In K. P. Jantke, editor, *Analogical and Inductive Inference, Lecture Notes in AI 397*, pages 18–44. Springer-Verlag, Berlin, 1989.
19. B. A. Ribeiro-Neto, A. H. F. Laender, and A. Soares da Silva. Extracting semistructured data through examples. In *Proceedings of the 1999 ACM International Conference on Information and Knowledge Management (CIKM'99)*, pages 94–101, 1999.
20. S. Soderland. Learning information extraction rules for semistructured and free text. *Machine Learning*, 34(1–3):233–272, 1999.

Querying Relational Databases without Explicit Joins[★]

Ramon Lawrence¹ and Ken Barker²

¹ University of Iowa, Iowa City, IA, USA
`ramon-lawrence@uiowa.edu`

² University of Calgary, Calgary, Alberta, Canada
`barker@cpsc.ucalgary.ca`

Abstract. Despite its benefits and wide-spread acceptance, SQL [5] is not a perfect query language. Although graphical tools for query construction mask some of the complexity, complex database schema challenge even experienced database users during query formulation because a user is responsible for mapping the semantics of their query to the structure of the database. In this work, we propose a semantic query language for graphically querying relational database systems that allows a user to query the database by semantics instead of structure. Database semantics are described using a global dictionary and semantic specifications that are combined to form an integrated, context view. Users query the semantic view by concept name, and the query processor translates semantic queries to SQL. This translation involves automatically determining attribute and relation mappings and join conditions.

1 Introduction

Despite changes in database size, complexity, and interoperability, SQL [5] has remained fundamentally unchanged. Although the complexity of SQL generation has been partially hidden by graphical design tools and more powerful programming languages, a SQL user is responsible for understanding the structure of a database schema, the names associated with schematic elements, and the semantics of the data stored. Query formulation involves mapping query semantics into the semantics of the database and then realizing those semantics by combining the appropriate database structures.

To simplify querying, we have implemented a new semantic query system. First, a database administrator semi-automatically produces a semantic view for querying using a global dictionary and a semantic specification called an X-Spec. A description of the capture process is in Section 2. Then, users graphically browse the integrated view to formulate queries. The query environment guides the user during query formulation allowing the user to unambiguously define the semantics of their query without explicitly referencing database structures or joins between relations. The query processor described in Section 3 is capable

[★] This research is sponsored by NSERC Research Grant (RGP-0105566) and TRILabs.

of automatically translating from the semantic context view to SQL queries by determining correct attribute and relation mappings and appropriate join conditions. The paper closes with related work and conclusions.

2 Semantic Capture Process

Before semantic querying can be performed on a relational database, the database administrator must explicitly define its semantics in a specification document. In this *capture process*, the database administrator (DBA) uses an automated tool to extract database schema information (including structure, types, sizes, names, and relationships) and saves the information into an XML [13] document called an X-Spec. An X-Spec provides information for semantic querying.

An example order database (given in Figure 1) is used throughout the paper. We denote attribute and relation names as *attribute_name* or *relation_name*, and semantic names as **semantic_name**.

Relations	Attributes
Categories	<u>CategoryID</u> , CategoryName
Customers	<u>CustomerID</u> , CompanyName
Employees	<u>EmployeeID</u> , LastName, FirstName, ReportsTo
OrderDetails	<u>OrderID</u> , <u>ProductID</u> , UnitPrice, Quantity
Orders	<u>OrderID</u> , CustomerID, EmployeeID, OrderDate, Shipvia
Products	<u>ProductID</u> , ProductName, SupplierID, CategoryID
Shippers	<u>ShipperID</u> , CompanyName
Suppliers	<u>SupplierID</u> , CompanyName

Fig. 1. Order Database Schema

2.1 Global Dictionary Definition

The foundation of the query system is the definition of a global term dictionary that provides terms to represent concept semantics. The dictionary is organized as a hierarchy of concept terms. Concept terms are related using 'IS-A' relationships for modeling generalization and specialization and 'HAS-A' relationships to construct component relationships. Each concept term has a name, an unique key, and an unique definition.

We have built a term dictionary starting from the top-level ontological categories proposed by Sowa [10], and evolve it using a set of rules. However, the exact terms and their placement is irrelevant. The dictionary can be modified to capture database semantics for a particular environment. The global dictionary is not a complete English language dictionary such as WordNet [8], so its size and complexity are reduced.

2.2 Semantic Names

A *semantic name* captures system-independent semantics of a relational schema element including contextual information by combining one or more dictionary terms. A semantic name is a **context** if it is associated with a relation and a **concept** if it is associated with an attribute. A context contains no data itself and is described using one or more concepts. Similar to an attribute in the relational model, a concept represents atomic or lowest-level semantics.

A semantic name consists of a context and concept portion. The **context portion** is one or more terms from the dictionary that describe the context of the schema element. Adjacent context terms are related by either IS-A (represented using a “,”) or HAS-A (represented using a “;”) relationships. The **concept portion** is a single dictionary term called a concept name and is only present if the semantic name is a concept (maps to an attribute). The formal specification of a semantic name is as follows:

$$\begin{aligned} \text{semantic\_name} &::= [\text{CT\_Term}] \mid [\text{CT\_Term}] \text{CN} \\ \text{CT\_Term} &::= \text{CT} \mid \text{CT} ; \text{CT\_Term} \mid \text{CT} , \text{CT\_Term} \\ \text{CT} &::= \langle \text{context term} \rangle, \text{CN} ::= \langle \text{concept name} \rangle \end{aligned}$$

A *dictionary term* is a single, unambiguous word or word phrase present in the standard term dictionary. Each term represents a unique semantic connotation of a given word phrase, so words with multiple definitions are represented as multiple terms in the dictionary. A *context term* is a dictionary term used in a semantic name that describes the context of the schema element associated with the semantic name. A *concept term* is a single dictionary term used in a semantic name that provides the lowest level semantic description of an attribute.

Definition. The *context closure* of a semantic name S_i denoted S_i^* is the set of semantic names produced by combining ordered subsets of the set of terms $T = \{T_1, T_2, \dots, T_N\}$ of S_i starting from T_1 . ■

Example If $S_i = [A; B; C]D$, $S_i^* = \{[A], [A; B], [A; B; C], [A; B; C]D\}$.

Overall, a semantic name is a combination of dictionary terms constructed by the DBA to represent schema element semantics. By combining terms, more complex semantics can be defined as compared to using terms in isolation.

2.3 X-Specs

The X-Spec is a database schema encoded in XML format and is organized in relational form with relations and attributes as basic elements. An X-Spec stores a relational database schema including keys, relationships, joins, and field semantics. Further, each relation and attribute in the X-Spec has an associated semantic name built using dictionary terms. Information on joins including their cardinality, join fields, and connecting tables is stored so that the query processor may identify which joins to apply during query formulation.

An X-Spec is constructed using a specification editor tool [6] during the capture process, where the semantics of schema elements are mapped to semantic names. X-Specs are constructed using XML because XML is an emerging semantic exchange standard. However, X-Specs may also be represented as formatted text files or structured binary files.

Type	Semantic Name	System Name
Relation	[Category]	Categories
Attribute	[Category] Id	CategoryID
Attribute	[Category] Name	CategoryName
Relation	[Customer]	Customers
Attribute	[Customer] Id	CustomerID
Attribute	[Customer] Name	CompanyName
Relation	[Employee]	Employees
Attribute	[Employee] Id	EmployeeID
Attribute	[Employee;Name] First Name	FirstName
Attribute	[Employee;Name] Last Name	LastName
Attribute	[Employee;Supervisor] Id	ReportsTo
Relation	[Order;Product]	OrderDetails
Attribute	[Order] Id	OrderID
Attribute	[Order;Product] Id	ProductID
Attribute	[Order;Product] Price	UnitPrice
Attribute	[Order;Product] Quantity	Quantity
Relation	[Order]	Orders
Attribute	[Order] Id	OrderID
Attribute	[Order;Customer] Id	CustomerID
Attribute	[Order;Employee] Id	EmployeeID
Attribute	[Order] Date	OrderDate
Attribute	[Order;Shipper] Id	Shipvia
Relation	[Product]	Products
Attribute	[Product] Id	ProductID
Attribute	[Product] Name	ProductName
Attribute	[Product;Supplier] Id	SupplierID
Attribute	[Product;Category] Id	CategoryID
Relation	[Shipper]	Shippers
Attribute	[Shipper] Id	ShipperID
Attribute	[Shipper] Name	ShipperName
Relation	[Supplier]	Suppliers
Attribute	[Supplier] Id	SupplierID
Attribute	[Supplier] Name	SupplierName

Fig. 2. Order Database Semantic Name Mappings

X-Spec information at the relation level stored in $X_T = (T_{SN}, T_Q, T_K, T_J)$ records consists of the relation semantic name T_{SN} , the SQL query T_Q used to produce the relation/view, T_K the set of keys for the relation, and T_J the set of joins for the relation. For a view T_Q is the SQL query executed to produce the view, or T_Q is the relation name for physical relations.

Attribute level information in $X_F = (F_{SN}, F_Q, F_T, F_S)$ stores the attribute semantic name F_{SN} , the SQL query used to produce the attribute F_Q , the attribute type (integer, string, etc.) F_T , and the attribute size in bytes F_S . F_Q is the system name for a physical attribute (e.g. [OrderDetails].OrderID) or a SQL statement used to construct a derived or calculated attribute.

The primary difference between an X-Spec and a relational schema is that each relation and attribute are assigned meaningful semantic names. Further, the model allows virtual constructs such as views and calculated attributes to be assigned semantic names. This allows a DBA to build more complex integrated views that consist of physical and virtually-derived constructs.

2.4 Semantic View Construction

An integration process formats the database X-Spec information into an integrated view of concepts by matching semantic names. Since a semantic name is itself a hierarchy of terms, the resulting integrated context view (CV) is a merger of all concepts across all attributes and relations in the database.

Definition. Define a *context view (CV)* as follows:

- If a semantic name S_i is in CV , then for any S_j in S_i^* , S_j is also in CV .
- For each semantic name S_i in CV , there exists a set of zero or more mappings that associate a schema element with S_i .
- A semantic name S_i can only occur in the CV once.

The integration process builds $CV = (CV_{SN}, CV_Q)$, where $CV_{SN} \in \{T_{SN} \cup F_{SN}\}^*$ and $CV_Q \in \{T_Q \cup F_Q\}$. That is, the CV is the union of the semantic names for all relations and attributes along with their associated mappings. The CV is a structurally-neutral, hierarchical representation of database concepts. The integrated view for the order database¹ is in Figure 3.

3 Query Processing

Users formulate queries by manipulating elements in the context view in a process called *querying by context*. The context view provides physical and logical access transparency. The user is not responsible for determining schema element mappings or joins between relations in the database.

A semantic query $Q = (Q_{SN}, Q_R, Q_O, Q_G, Q_C)$ where Q_{SN} is the set of semantic names selected in the query, Q_R is the set of relationship conditions between semantic names, Q_O is the set of ordering criteria for the row results, Q_G is the set of grouping criteria, and Q_C is the set of selection criteria. Ordering and selection criteria are specified using semantic names rather than system names. Thus, for a given Q , a translation is required to a SQL query of the form $SQ = (SQ_T, SQ_F, SQ_J, SQ_O, SQ_G, SQ_C)$, that contains the required relation set (SQ_T), attribute set (SQ_F), joins (SQ_J), ordering criteria (SQ_O), grouping criteria (SQ_G), and selection criteria (SQ_C).

The query processor performs this translation during **dynamic view creation** that extracts relevant data from the database and requires the system:

- **Enumerate Semantic Names** - required by the user.
- **Determine Relevant Attributes and Relations** - for each semantic name mapping.
- **Determine Join Conditions** - to connect the relations.
- **Generate and Execute SQL Queries** - created in the previous steps.

¹ For simplicity, only the term at a given level is displayed. For example, the term in bold has full semantic name [Order;Product] Id as Id is under the [Product] and [Order] contexts.

Integrated View Term	Data Source Mappings (not visible to user)
V (view root)	N/A
- [Category]	Categories
- Id	Categories.CategoryID
- Name	Categories.CategoryName
- [Customer]	Customers
- Id	Customers.CustomerID
- Name	Customers.CompanyName
- [Employee]	Employees
- Id	Employees.EmployeeID
- [Name]	
- First Name	Employees.FirstName
- Last Name	Employees.LastName
- [Supervisor]	
- Id	Employees.ReportsTo
- [Order]	Orders
- Id	Orders.OrderID, OrderDetails.OrderID
- Date	Orders.OrderDate
- [Customer]	
- Id	Orders.CustomerID
- [Employee]	
- Id	Orders.EmployeeID
- [Product]	OrderDetails
- Id	OrderDetails.ProductID
- Price	OrderDetails.UnitPrice
- Quantity	OrderDetails.Quantity
- [Shipper]	
- Id	Orders.Shipvia
- [Product]	Products
- Id	Products.ProductID
- Name	Products.ProductName
- [Supplier]	
- Id	Products.SupplierID
- [Category]	
- Id	Products.CategoryID
- [Shipper]	Shippers
- Id	Shippers.ShipperID
- Name	Shippers.ShipperName
- [Supplier]	Suppliers
- Id	Suppliers.SupplierID
- Name	Suppliers.SupplierName

Fig. 3. Integrated View of Order Database

3.1 Enumerating Semantic Names

Semantic names are selected by the user for display in the final result (projection) or for specifying selection criteria. In some cases, a real-world concept may be represented multiple times in the database and possibly with different semantic names. The first task for the user is to determine the correct semantic name for querying. For example, in the example database there are two semantic names that map to the concept of a “Shipper ID”: *ShipperID* ([**Shipper**] Id) in *Shippers* and *Shipvia* ([**Order**; **Shipper**] Id) in *Orders*. The choice of semantic name depends on the query requirements. Selecting [**Shipper**] Id will select all shipper ids whether or not they have transported an order, whereas [**Order**; **Shipper**] Id only returns shippers who have sent orders.

The user selects the concepts out of the integrated view they are interested in for their query, and may specify selection criteria, grouping criteria, and ordering

criteria using semantic names. The user also has control over the final display characteristics of each concept selected. Since the integrated view is organized hierarchically by concept, knowledge discovery is considerably simplified when browsing the integrated view.

3.2 Attribute and Relation Selection

The query system determines which relations and attributes to access in the data source based on the semantic names chosen by the user. In most cases, a semantic name in the integrated view has only one mapping to a physical attribute. However, in special cases, especially when considering key attributes, a semantic name may map to several attributes. The choice of attribute (and its corresponding relation) may affect the semantics of the query. An overview of the attribute and relation selection algorithm is:

1. For each semantic name F_{SN} with only one mapping, add attribute to SQ_F and corresponding table to SQ_T .
2. For each semantic name F_{SN} with multiple mappings all in one relation², add each attribute mapping to SQ_F and the corresponding table to SQ_T .
3. For any semantic names remaining with ambiguous mappings, select an attribute mapping to SQ_F for the semantic name if its corresponding table is already in SQ_T .
4. For any remaining semantic names with ambiguous mappings, select the mapping based on shortest join paths to current relations in SQ_T (or other heuristics). This new mapping adds a new relation to SQ_T , so goto step 3.

In practice, the attribute mapping algorithm rarely uses the heuristics on join cost calculations in Step 4 because typically only key attributes will have multiple semantic name mappings that result from database normalization. When the user selects any attributes with unique mappings, the attribute selection algorithm has a starting set of relations to build a query from and determine which key mapping instance(s) to select based on the general heuristic of not introducing new joins into the query.

After completion of the attribute selection algorithm, the set of semantic names Q_{SN} in Q has been translated to the set of attributes SQ_F and relations SQ_T for the SQL query. Further, the system can replace the semantic names present in Q_O , Q_G , and Q_C for ordering, grouping, and selection criteria with system attribute and relation names for SQ_O , SQ_G , and SQ_C based on the discovered mappings.

3.3 Determining Join Conditions

Given a set of attributes and relations to access in the query, the query system must determine a set of join conditions between the relations. It is important

² This may occur when the database is not normalized. The system queries non-normalized records and normalizes the data when it is presented to the user.

to isolate the user from join construction while choosing appropriate joins to preserve the query semantics.

There is no concept of a “join” in the integrated view because it does not specify a structure for data representation. However, an equivalent operation is a context merger. A *context merger* is the combination of two contexts by applying a relationship condition. A relationship condition may be a join between relations if a join exists between the contexts or a cross-product if it does not.

In most cases³, the user never directly specifies the relationship condition between two contexts in the integrated view, so the system must determine how to relate the two contexts. Since the user is able to query on any semantic name in the integrated view, the semantic concepts chosen may be related by being in the same relation, by joins between relations, or not related at all.

The system determines the joins to apply using join graphs. Define a *join graph* as an undirected graph where each node corresponds to a relation in the database, and there is a link from node N_i to node N_j if there is a join between the corresponding two relations. A *join path* is a sequence of one or more joins interconnecting two nodes (relations), and a *join tree* is a set of one or more joins interconnecting two or more nodes. Assume without loss of generality⁴ that the join graph is connected. The join graph for the order database is in Figure 4.

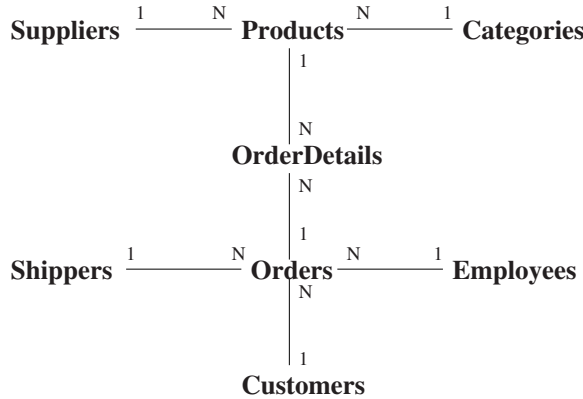


Fig. 4. Join Graph for Order Database

Lemma 1. *If a join graph is acyclic, there exists only one join path between any two nodes.*

Lemma 2. *If a join graph is acyclic, there exists only one join tree for any node subset.*

³ A user can override default system relationship conditions if the database contains multiple, ambiguous joins.

⁴ Otherwise, we apply the algorithm to each connected subset and connect them using a cross-product.

Lemmas 1 and 2 follow directly from the properties of acyclic graphs. Importantly, if the join graph for a database is acyclic, there exists only one possible join tree for any of its relations. This implies that the query system does not have any decisions involving which joins to apply, as it must only identify which joins are required to connect the required tables by constructing the join tree.

From this result, it is possible to construct an algorithm that builds a matrix M where entry $M[N_i, N_j]$ is the shortest join path between any pair of nodes N_i and N_j . By combining join paths, the query system can identify all the joins required to combine relations by constructing the only possible join tree.

Theorem 1. *Given a matrix M , which stores shortest join paths for an acyclic join graph, and a set of relations T to join, a join tree can be constructed by choosing any relation T_i from T and combining the join paths in $M[N_i, N_1]$, $M[N_i, N_2]$, ..., $M[N_i, N_n]$ where $N_1, N_2, \dots, N_n$ are the nodes corresponding to the set of relations T .*

Proof. Proof by contradiction. Since the graph is connected, the matrix entries $M[N_i, N_1]$, $M[N_i, N_2]$, ..., $M[N_i, N_n]$ represent join paths from N_i to all other nodes in the subset. Assume a join tree is not constructed. Thus, there is no path between some two nodes N_j and N_k where $j \neq k$. However, there is a path from node N_i to N_j and from node N_i to N_k . Combining these paths results in a path from N_j to N_k . Thus all nodes are connected with the join tree, and it is the only possible join tree as per Lemma 2. ■

For the general case of a cyclic join graph, we define a depth-first search based algorithm that constructs a matrix M containing all join paths with no repeated links ranked by desirability. Join ranking is based on join properties such as cardinality, participation, and if they are lossy joins. Although the algorithm ranks join paths, in practice, only acyclic graphs can have their joins automatically determined without further user input by applying Theorem 1. For a cyclic graph, there will be multiple possible join trees each of which are semantically valid depending on the query semantics. The system cannot select a join tree without more knowledge about the intended query, although it is able to propose the “best” join tree based on the ranking performed. Thus, we define a graphical interface to the query model that allows the user to browse the integrated view, discover concept interrelationships, and more precisely define their query semantics which allows the system to uniquely determine the join tree required.

3.4 Graphically Guiding the User during Query Generation

The query system currently described is unable to determine join trees without further user input for cyclic join graphs. The challenge is to extract the required information from the user while still maintaining physical and logical query transparency as much as possible. The system allows the user to browse the context view to discover interrelationships between concepts. Some concepts are already inherently related by virtue of their hierarchical relationship. For example, the [Order;Product] Id is part of [Order] information. This relationship is explicitly captured in the integrated view by virtue of the product information being nested under the order information in the hierarchy. However,

other interrelationships cannot be captured using the strict hierarchy implicit in the semantic names. For instance, the join conditions relating the individual concepts are largely hidden to the user, even though these join conditions are automatically inserted by the query system during query execution. To make these interrelationships more apparent, the system automatically displays them. If a given semantic name (node) in the integrated view is actually a foreign key to another context (table) then when the user clicks on this concept, the attributes of the linked context are displayed.

For example, the field *EmployeeID* ([Order;Employee] Id) in *Orders* is a foreign key from *Orders* to *Employees*. When the user clicks on this semantic name, the system performs the join to the *Employees* table and displays the semantic names of the fields of *Employees* which can be added to the query.⁵

This approach has several benefits. First, it reduces the semantic burden on the user by automatically displaying concept interrelationships. Further, it reduces the query generation complexity for the system. By explicitly displaying the join information and associated attributes, the system now has an unambiguous reference from the user on which attributes to use, from what relations, and the corresponding join condition (attribute) to use to relate the two different contexts. Further, by allowing this virtual linking of concepts, the user is now able to express recursive relationships and queries on the integrated view.

To formalize the discussion, define a *context tree* as a tree of nodes where each node is a single semantic name, and at every node we have a list of attributes accessed from that context. The root node of the tree is *V*, the view root. There is a link from a parent node (context) in the tree to a child node, if the child node is a subcontext of the parent.

For example, let *OrderDetails* also have a foreign key *EmployeeID* to *Employees*. This produces a cycle in the join graph. Now, consider the query requiring the [Employee] Name, [Order] Id, and [Order;Product] Amount. The context tree for this query is given in Figure 5a. In the cyclic join graph, there are multiple possible join trees to connect these contexts. In general, if there is more than one child node of *V* and the graph is cyclic, there will be multiple join trees that relate the contexts.

A different query can be derived by the user explicitly following the link to employee information through [Order;Employee] Id (Figure 5b). The user can specify another distinct join by selecting [Employee] Name after traversing the foreign key in *OrderDetails* to *Employee*. In this way, the user uniquely identifies the joins required by the system by graphically browsing the context view.⁶

Linking of contexts allows the system to handle recursive queries. *Employees* has a foreign key *ReportsTo* that links to *Employees* on *EmployeeId* in order to model that an employee can have a supervisor. The system can answer a query like: “who is an employee’s supervisor’s supervisor?” by allowing the user to graphically traverse the context view (see Figure 6).

⁵ If multiple joins are possible on a given foreign key, the system allows the user to select which context (relation) they are interested in connecting to.

⁶ Advanced queries involving right or left joins can be directly specified by the user by displaying the current context tree and allowing users to manipulate it.

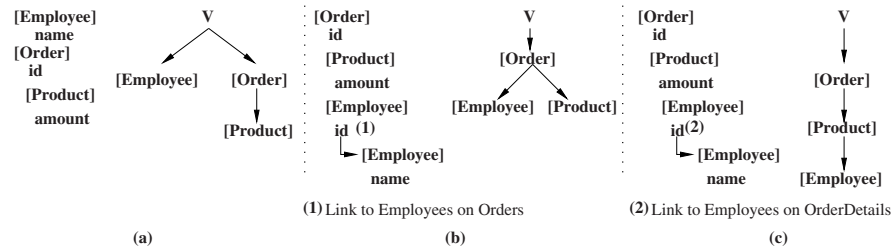


Fig. 5. Example Queries

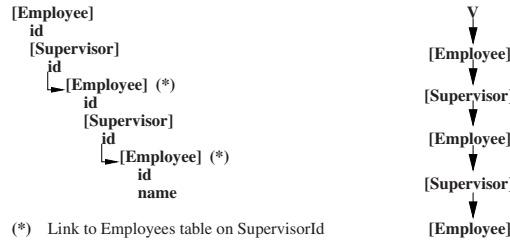


Fig. 6. Recursive Supervisor Query

3.5 Generation and Execution of SQL Queries

Given the set of database attributes and relations to access and a set of joins to apply, it is straightforward to construct a SQL select-project-join query. After building a SQL query string, the query is transmitted to the database management system for execution. Although there are multiple open standards and proprietary protocols for each database and environment, the architecture is designed to utilize the ubiquity of the ODBC standard to access all major database systems. Results returned from each ODBC query are then processed by the client to perform global level formatting. Two example queries and their associated SQL mappings are below.

- Example 1 a) The user requires all customer names ([Customer] Name).
b) The user requires all customer names with orders ([Order;Customer] Name).

(a)	(b)
Select CompanyName	Select CompanyName
From Customers	From Customers, Orders
	Where Customers.CustomerID = Orders.CustomerID

Execution 1 a) Retrieves customers whether or not they have an order. No context merging is required. b) The customer name is retrieved from the Customers relation. The query processor performs a context merger by combining the [Order] context with the [Customer] context. A join is required between Orders and Customers to merge the two contexts.

4 Related Work

SQL provides an efficient and structured way for accessing relational data. However, specifying complex SQL queries with numerous join conditions and subqueries is too complex for most users [1]. Further, developing SQL queries requires knowledge of both the structure and semantics of the database. Unfortunately, database semantics are not always immediately apparent from the database schema, and mapping the required query semantics into a SQL query on database structure is often complex. There are graphical query tools [2,11] to aid in the formulation of SQL queries and proposed extensions to the SQL language exist [12]. Many commercial databases use similar tools to aid the user in query construction. However, at the lowest level, a user is still responsible for mapping the semantics of their query into a structural representation suitable for the database.

Query by example or QBE [14] was proposed to reduce this semantic burden on users. By providing a table-based query environment and a simple query syntax, QBE demonstrated that some of the burden of accessing relational databases can be mitigated by using advanced user interfaces that capture the semantics of a user's query, and automatically translate them to relational queries. Query by context (QBC) differs from query by example as QBC does not have a structural basis for the query. QBE's foundation rests on a table structure. Although QBE is effective at hiding much of the query complexity, QBC is a further advancement by hiding the structure as well. Further, this structural neutrality allows queries to be unaffected by database structural changes as long as the same semantic concepts are present, and will allow QBC to be extended to object-oriented databases and multidatabases.

Another methodology for simplifying query complexity is to hide the structure of the database by allowing some form of querying by natural language. Systems [4,9] have been developed that allow users to query by word phrases, but these systems are limited if they do not allow the user to precisely define the exact data returned. Unlike a SQL query which is deterministic and precise, query by word systems that simplify query formulation by ignoring structure, sacrifice query precision.

In a general environment, a query system must isolate the user from structure and system details while at the same time provide a query language powerful enough to produce precise, formatted results. SemQL [7] provides semantic querying using semantic networks and synonym sets from WordNet [8]. Although their approach is similar to ours, using a large online dictionary such as WordNet increases the complexity of matching word semantics. Also, since no integrated view is produced, it is not clear to the user which concepts are present in the databases to be queried. Our approach improves on SemQL by providing a condensed term dictionary, an integrated view to convey database semantics to the user, and a systematic method for SQL generation.

A different approach to relational querying involves using an intelligent query processor that directs the user during query formulation. Kaleidoscope [3] provides a logic-based interface that allows a user to formulate queries using a restricted natural language syntax. As the user builds a query, Kaleidoscope

uses rule-based and constraint knowledge to ensure a well-formed query, thus minimizing the chances for errors and easing the cognitive burden on the user.

A decent measure of user query performance is a count of the number of symbols or constructs required to formulate a query. Querying by context requires limited user input and interaction to formulate a query. For all systems, a user must select the concepts they are interested in. This process is performed in querying by context by allowing a user to graphically browse the context view. By organizing concepts by semantics instead of structure and using meaningful semantic names, the burden of concept selection and discovery is significantly reduced. Further, QBC attempts to minimize the user's specification of concept interrelationships. Concept interrelationships are automatically determined if possible, or can be easily specified by the user by traversing the integrated view. The concept of linking contexts by traversing the integrated view is similar to links connecting HTML pages. Even when the user must specify interrelationships, this is a simpler task than determining joins to connect physical relations as there will often be fewer of them. Overall, user query generation complexity is reduced by organizing and naming contexts and concepts semantically, isolating the user from database structures and naming (and the associated semantic to structural translation), and by providing a graphical query environment that allows the user to build queries by manipulating the simple semantic notions of concepts, contexts, and relationships between contexts.

5 Future Work and Conclusions

In this paper, we have demonstrated that by utilizing a global dictionary and semantic specifications an integrated, context view of a database can be constructed. This semantic view is easier for the user to query, as they are isolated from structural and naming considerations. We have discussed an automatic system that maps semantic queries to SQL and presents the results to the user. The query system is capable of handling complex join constructs and choosing the appropriate attributes, relations, and join conditions to preserve user query semantics. A graphical query environment is developed to allow the user to more formally define the semantics of their query without explicit knowledge of the structure and interrelationships of database attributes and relations. The query system described has been implemented in a software package called Unity [6].

Future work involves expanding the query processor to handle updates, and experimental comparisons of query by context with SQL, natural language querying systems, and other graphical query systems.

References

1. J. Bell and L. Rowe. Human Factors Evaluation of a Textual, Graphical, and Natural Language Query Interfaces. Technical Report ERL-90-12, University of California, Berkeley, February, 1990.
2. T. Catarci and G. Santucci. Query by Diagram: A Graphical Environment for Querying Databases. *SIGMOD Record*, 23(2):515–515, June 1994.

3. S. Cha and G. Wiederhold. Kaleidoscope Data Model for An English-like Query Language. In *17th International Conference on Very Large Data Bases*, pages 351–361. Morgan Kaufmann, 1991.
4. W. Cohen. Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity. *SIGMOD Record*, 27(2):201–212, 1998.
5. C. Date. *The SQL standard*. Addison Wesley, Reading, US, third edition, 1994.
6. R. Lawrence and K. Barker. Unity - A Database Integration Tool. Technical Report 2000-664-16, Dept. of Computer Science, University of Calgary, July 2000.
7. J. Lee and D. Baik. SemQL: A Semantic Query Language for Multidatabase Systems. In *Proceedings of the 8th International Conference on Information Knowledge Management (CIKM'99)*, pages 259–266, Kansas City, MO, November 1999.
8. G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five Papers on WordNet. Technical Report CSL Report 43, Princeton University, 1990.
9. W. Ogden and S. Brooks. Query Languages for the Casual User: Exploring the ground between Formal and Natural Languages. In *Proceedings of the Annual Meeting of the Computer Human Interaction of the ACM*, pages 161–165, 1983.
10. J. Sowa. Top-level ontological categories. *International Journal of Human-Computer Studies*, 43(5):669–685, 1995.
11. M. Stonebraker, J. Chen, N. Nathan, C. Parson, A. Su, and J. Wu. Tioga: A Database-Oriented Visualization Tool. In *Proceedings of the Visualization '93 Conference*, pages 86–93. IEEE Computer Society Press, October 1993.
12. G. Vossen and J. Yacabucci. An extension of the database language SQL to capture more relational concepts. *SIGMOD Record*, 17(4):70–78, December 1988.
13. W3C. Extensible Markup Language (XML) 1.0. Technical Report <http://www.w3.org/XML/>, February 1998.
14. M. Zloof. Query-by-Example: a data base language. *IBM Systems Journal*, 16(4):324–343, 1977.

NF-SS: A Normal Form for Semistructured Schema

Xiaoying Wu¹, Tok Wang Ling¹, Sin Yeung Lee¹, Mong Li Lee¹, and
Gillian Dobbie²

¹School of Computing, National University of Singapore, Singapore
{wuxiaoy1, lingtw, jlee, leeml}@comp.nus.edu.sg

²Dept of Computer Science, University of Auckland, New Zealand
gill@cs.auckland.ac.nz

Abstract. Semistructured data is becoming increasingly important for web applications with the development of XML and related technologies. Designing a “good” semistructured database is crucial to prevent data redundancy, inconsistency and undesirable updating anomalies. However, unlike relational databases, there is no normalization theory to facilitate the design of good semistructured databases. In this paper, we introduce the notion of a semistructured schema and identify the various anomalies that may occur in such a schema. A Normal Form for Semistructured Schemata, NF-SS, is proposed. A semistructured schema in NF-SS guarantees minimal redundancy and hence no undesirable updating anomalies for the associated semistructured databases. Furthermore, a semistructured schema in NF-SS gives a more reasonable representation of real world semantics. We develop an iterative algorithm based on a set of heuristic rules to restructure a semistructured schema into a normal form. These design methods also provide insights into the normalization task for semistructured databases.

1. Introduction

Semistructured data plays a crucial role in the new Internet applications ranging from electronic commerce to web site management to digital government. The emergence of XML (eXtended Markup Language) [2] as the likely standard for representing and exchanging data on the web has confirmed the central role of semistructured data. Many information providers have published their databases on the web as semistructured data, and others are developing repositories for new applications. As with traditional databases, data redundancy and inconsistency may occur in a semistructured database if its schema is not designed properly. Thus, it is important to provide guidelines for designing “good” semistructured databases.

Unlike the relational model where normalization theory is used to decide whether a set of relations is a good design for a given database, there is no normalization theory defined for semistructured data to determine whether a semistructured database has been well-designed. This is a major problem in the field of semistructured data research.

Normal forms defined for relational databases, either 3NF, BCNF, 4NF and 5NF for the flat relational model, or nested relations like NNF [8] and NF-NR [7], are not directly applicable to semistructured data for the following reasons. First, the semistructured data model is richer and more complex than the flat relational data model. For example, XML incorporates cardinality constraints that are not found in the relational data model. Second, a semistructured data instance whose structure is embedded together with the data, is required to *conform to* its schema. Hence there is no regular structure in semistructured data instances. Third, unlike comparing values of atomic types, it is a nontrivial task to directly compare values of hierarchically structured data. The notion of value equality for semistructured data has to be defined. Fourth, dependency constraints (such as functional and multi-valued dependencies) used in traditional design approaches are not directly applicable to the semistructured data model.

In this paper, we propose NF-SS, a normal form for semistructured schema. We use XML as a data model to represent semistructured data and enrich the model with schema and integrity constraints such as dependency and key constraints. We will show that a semistructured schema in NF-SS not only reduces redundancy and undesirable updating anomalies for the complying semistructured databases, but also captures the set of semantic connections among objects and attributes that exist in the real world. We will propose a set of heuristic restructuring rules and use them in an algorithm to obtain a NF-SS schema.

The rest of paper is organized as follows. Section 2 gives a motivating example and identifies the anomalies that may occur in semistructured databases. Section 3 defines NF-SS, a normal form for semistructured schema. Section 4 presents a set of heuristic rules to restructure a semistructured schema. An algorithm that iteratively transforms a semistructured schema into NF-SS is also given. Section 5 discusses some related works and we conclude in Section 6 with directions for future work.

2. Motivating Example

In this section, we will give an example to demonstrate that without some guide to design schemas, it is easy to produce semistructured schemas that contain redundancy and unnatural representation of real word semantics.

Example 2.1 Consider the XML DTD in Figure 2.1. A graphical representation of the DTD is shown in Figure 2.2(a). A database instance that conforms to this DTD is given in Figure 2.3. Note that this database is not well designed because it contains data redundancy: the *name* and *age* of a *student* will be repeated for each additional *course* he/she takes. Similar to traditional databases, we can identify three kinds of update anomalies in a badly designed semistructured database: *insertion anomaly*, *rewriting anomaly* and *deletion anomaly* (see [9] for more details). This redundancy can be avoided if the database is designed according to the schema shown in Figure 2.2(b), where *student's* information, including name and age, is referenced rather than nested under *course*.

```
<!ELEMENT department (course+)>
<!--ATTLIST
  department
  name ID #REQUIRED-->
<!ELEMENT course
  (students*)>
<!--ATTLIST
  course
  cid ID #REQUIRED
  title CDATA #IMPLIED-->
<!ELEMENT student
  (grade?)>
<!--ATTLIST
  student
  sid ID #REQUIRED
  name CDATA #REQUIRED
  age CDATA #IMPLIED-->
<!ELEMENT grade
  (#PCDATA)>
```

Fig. 2.1: Example of a DTD.

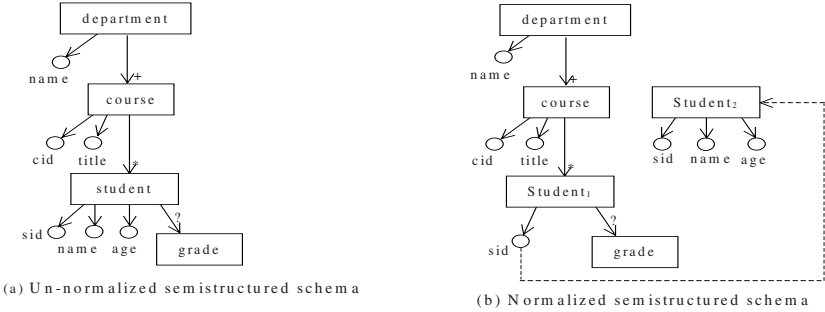


Fig. 2.2: Graphical representations for the DTD in Figure 2.1.

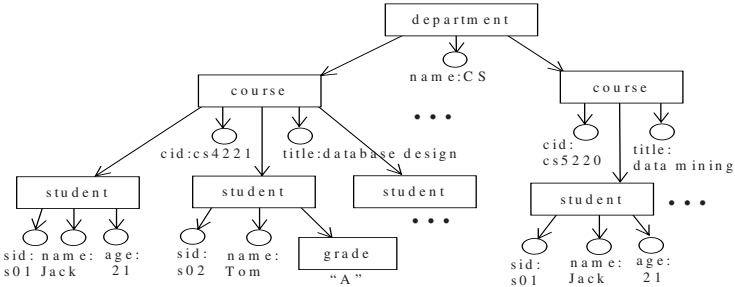


Fig. 2.3: A semistructured database instance that conforms to the schema in Figure 2.2(a).

3. A Normal Form for Semistructured Schema (NF-SS)

In this section we will give the formal definition of a semistructured schema as well as the concept of data tree that conforms to its schema. We will also introduce the concept of Extended Functional Dependency (EFD) and key constraints defined over semistructured schema. Finally, we will present the definition of NF-SS.

3.1 Semistructured Schema (D) and Data Tree (T)

Definition 3.1: A *semistructured schema* $D = (E, A, B, P, R, r)$ where

- E is a finite set of *object types* in D .
- A is a finite set of *attributes*, disjoint from E .
- B is a set of basic domain type like *string*, *integer*, *Boolean* etc.
- P is a function from E to *object type definition*, which is an expression $o_1^{w_1}, \dots, o_k^{w_k}$, where O_i is either a distinct object type in E or a basic domain type in B . Each W_i is a symbol in $\{*, +, ?, 1\}$ called *multiplicity*.
- R is a function from E to the power set of A . If $a \in R(o)$, then a is defined for o .
- $r \in E$ and is called the *object type of the root*.

We denote an object type o in an schema D with attributes $a_1, \dots, a_m$ and sequence of sub-object type $o_1, \dots, o_n$ by $o(a_1, \dots, a_m; o_1, \dots, o_n)$. D can be represented graphically as a tree with labeled rectangles and circles denoting object types and attributes respectively. Any multiplicity is indicated on the edges between object types.

Example 3.1 The schema in Figure 2.2(a) can be represented as $D=(E, B, A, P, R, r)$ where $E = \{\text{department, course, student, grade}\}$ $A = \{\text{cid, title, sid, name, age}\}$

$P(\text{department}) = \text{course}^+$	$R(\text{department}) = \{\text{name}\}$
$P(\text{course}) = \text{student}^*$	$R(\text{course}) = \{\text{cid, title}\}$
$P(\text{student}) = \text{grade}$	$R(\text{student}) = \{\text{sid, name, age}\}$
$P(\text{grade}) = \text{string}$	$r = \text{department}$

A semistructured database instance such as an XML document is usually modeled as a labeled tree. We define such a data tree as an image of a given semistructured schema as follows.

Definition 3.2: A *data tree* T w.r.t. a semistructured schema $D = (E, A, B, P, R, r)$ is defined to be a tree $T = (V, \text{lab}, \text{obj}, \text{att}, \text{val}, \text{root})$ where

- V is a finite set of nodes.
- lab is a labeling function that mapping each node in V to $E \cup A \cup B$.
- obj is a partial function from V to a sequence of nodes in V such that for any $v \in V$, $\text{obj}(v)$ is defined iff $\text{lab}(v)=o$, and $o \in E$; moreover, if $\text{obj}(v)=\langle v_1, \dots, v_n \rangle$, then $\langle \text{lab}(v_1), \dots, \text{lab}(v_n) \rangle$ must be in $\langle o_1^{w_1}, \dots, o_n^{w_n} \rangle$ defined by $P(o)$; and for $o_i^{w_i}$ in $P(o)$, the number of children of v that is labeled O_i as is restricted as follows:
 $w_i=1$: exactly one child is labeled O_i ; $w_i=?$: at most one child is labeled O_i ;
 $w_i=+$: at least one child is labeled O_i ; $w_i=*$: no restriction;
- att is a partial function from $V \times A$ to V such that for any $v \in V$ and $a \in A$, $\text{att}(v,a)=v'$ is defined iff $\text{lab}(v)=o$, $\text{lab}(v')=a$, $o \in E$, $v' \in V$ and $a \in R(o)$.
- val is a partial function from V to atomic values such that for any node $v \in V$, $\text{val}(v)$ is an atomic value iff $\text{lab}(v)=s$, $s \in B$ or $s \in A$.
- root is a distinguished node in V , $\text{lab}(\text{root})=r$, and is called the root of T .

Example 3.2 The data tree in Figure 2.3 is an instance of the schema in Figure 2.2(a). The labeled rectangles and circles denote object nodes and attribute nodes

respectively. In the department object node, *obj* defines its sub-objects such as course “cs4221” and teacher “t1234” etc., while *att* defines its attribute nodes labeled with name, which is assigned value “cs” by *val*.

For any node n in a tree, either a schema tree D or a data tree T , there exists a unique path starting from the root to n , denoted as $Path_D(n)$ or $Path_T(n)$ accordingly. Their formal definitions are provided in [10], and their notation makes use of XPath expression [4].

Since data tree T is an image of D , given a node $n \in E \cup A$ in D , there maybe many nodes in T whose path in T is equal to n 's path in D , we call such set of nodes as target set of n in T . Formally, it is defined as follows:

Definition 3.5: Let $T=(V, lab, obj, att, val, root)$ be a data tree of schema $D=(E, A, B, P, R, r)$. $n \in E \cup A$. The *target set* of n in T , denoted as $T[n]$, is $\{v: v \in V, Path_T(v)=Path_D(n)\}$

Example 3.3: Refer to the schema in Figure 2.2(a), the path for object type *student* is */department/course/student*. For its data tree T in Figure 2.3, the path for student “s01” is */department/course/student*; the target set $T[student]$ includes nodes of *students* with *sid* “s01” and “s02”.

3.2 Extended Functional Dependency (EFD) and Key Constraints

Extended Functional Dependency (EFD)

Before we give the definition for EFD, we first define some terminology that is fundamental to the semantics of EFD.

First, we provide the notion of equality of two nodes in data trees. Intuitively, two nodes n_1, n_2 are *value equal* denoted by $n_1 =_v n_2$, if they have the same label and either they have the same atomic value (when they are value or attribute nodes) or their children are pair-wise *value equal* (when they are object nodes).

Example 3.4 In Figure 2.3, the leftmost *student* node (*sid* is “s01”) and the rightmost *student* node (*sid* is “s01”) are value equal because they have the same label tag (*student*) and all their children $\langle sid, name, age \rangle$ are pairwise value equal.

In relational databases, when two tuples agree on a set of attributes X , this implies that their projections on X are equal. Similarly, two data trees T_1 and T_2 that agree on an object type or attribute X implies that there exists nodes in their target sets of X in T_1 and T_2 respectively satisfying value equality.

Definition 3.6: Let T_1 and T_2 be two data trees that are images of schema $D=(E, A, B, P, R, r)$. Let $X \in E \cup A$. T_1 and T_2 *agree on* X , denoted as $T_1 =_X T_2$ iff the following condition is hold: $\exists t_1 \in T_1[X], t_2 \in T_2[X]$, such that $(t_1 =_v t_2)$

Note that we do not require that the above two target sets to satisfy set equality, since it is possible that there are no such nodes in a data tree.

Definition 3.7: Let $D=(E, A, B, P, R, r)$ be a semistructured schema, let $X \subseteq E \cup A$ and $Y \in E \cup A$. Y is extended functionally dependent on X , is denoted as $X \Rightarrow Y$. Let S

denotes a set of data trees that are images of D , S satisfies $X \Rightarrow Y$, iff for any data trees T_1, T_2 in S , if they agree on every component in X , then they will agree on Y . That is,

$$\forall T_1, T_2 \in S((\forall x \in X, T_1 =_x T_2) \mapsto T_1 =_y T_2).^1$$

In the above definition, we write the EFD as $X \Rightarrow Y$ if Y is an attribute of an object type or a single valued object type. If there exists an $X' \subset X$ such that $X' \Rightarrow Y$, then the EFD $X \Rightarrow Y$ is called *partial* EFD; otherwise we say that $X \Rightarrow Y$ is a *full* EFD. If $Y \subseteq X$, then $X \Rightarrow Y$ is a *trivial* EFD. $X \Rightarrow Y$ is said to be *coherent* iff $/X/Y$ is a path in D ; otherwise it is called an incoherent EFD. If there exists $Z \in E \cup A$, such that $X \Rightarrow Y$ and $Y \Rightarrow Z$ and $Y \not\Rightarrow X$, then Z is transitively extended functionally dependent on X via Z . From the definition of partial EFD and transitive EFD as well as inference rules for EFD, it is easy to conclude that partial EFD is a special case of transitive EFD. We use the following notation for an EFD:

$$O_1[@X_1], \dots, O_i[@X_i], \dots, O_{n-1}[@X_{n-1}] \Rightarrow O_n[@X_n]$$

where O_i is an object type in D , X_i is a set of O_i 's attributes that participate in the dependency. Note that the notation makes use of $Xpath[4]$ expressions.

Example 3.5 Consider the schema in Figure 2.2(a). We have the following two EFDs:
 (1) $course[@cid], student[@sid] \Rightarrow student[@name]$
 (2) $student[@sid] \Rightarrow student[@name]$
 where (1) is a partial EFD while (2) is a full EFD since a *student*'s name is fully determined by *sid*.

An incoherent EFD introduces a path that is not expressed in the schema's structure. We say that the existence of incoherent EFD leads to *path anomaly* in schema that indicates an unintuitive grouping of objects and attributes in schema's structure. In such cases, the schema does not adequately reflect the real world semantic relationships and will lead to data redundancy and data retrieval overheads.

Example 3.6 Consider the schema D shown in Figure 3.1(a) that intends to be a schedule for teachers lecturing courses. Assume the specified EFD is $teacher[@tid], time[@day, @hour] \Rightarrow subject[@cid]$. It is an incoherent EFD, since $/teacher/time/subject$ is not a path in the schema. Therefore, there exists a path anomaly. Such anomaly can be avoided if we promote *time* to be child of *teacher*, and move *ClassRoom* and *subject* under *time*, as shown in Figure 3.1(b).

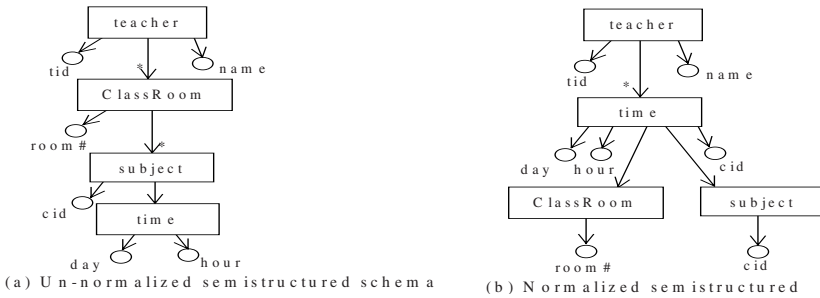


Fig. 3.1: Schema for Example 3.6 illustrating path anomaly

¹ The inference rules for EFD is given in [10]

Example 3.7 In Figure 2.2(a), *age* is transitively dependent on *course* via *student* given the constraints: (1) $\text{course}[\text{@cid}] \Rightarrow \text{student}[\text{@sid}]$
 (2) $\text{student}[\text{@sid}] \rightarrow \text{student}[\text{@age}]$ and
 $\text{student}[\text{@sid}] \not\rightarrow \text{course}[\text{@cid}]$

Theorem 3.1: Let $D = (E, A, B, P, R, r)$ be a semistructured schema, $X, Y, Z \subseteq E \cup A$. If Z is transitively dependent on X via Y , then there exists a data tree of D in which a *rewriting anomaly* arises when the values of Z are updated.

We will illustrate the above theorem with the following example. A detailed proof is given in [10].

Example 3.8 Refer to Figure 2.3. As aforementioned, *age* is transitively dependent on *course* via *student*. The age information for student with *sid* equal “s01” is repeated for two courses in which he is enrolled. If we update his age information under the course “cs4221”, then it can cause inconsistency unless the update is also propagated to his age information under the course “cs5220”.

Key Constraints

Key plays an important part in database design [1,9]. Unlike previous proposal that define key constraints for semistructured data in the absence of schema [3], we want to define the notion of key based on semistructured schema and EFD. This allows us to derive key constraints more easily.

Definition 3.8: Let $O(a_1, \dots, a_m; o_1, \dots, o_n)$ be a complex object type in semistructured schema, and K is a non-empty subset of $\{a_1, \dots, a_m, o_1, \dots, o_n\}$, where $\{o_1, \dots, o_n\} \subseteq \{o_1, \dots, o_n\}$, and are single-valued atomic sub-object types. The *key* of O is defined as:

Case 1: O is the root of D . K is a *candidate key* of O iff $K \rightarrow O$ and there is no proper subset K_I of K , such that $K_I \rightarrow O$.

Case 2: O is at some level in D . Let $\text{Path}_D(O) = /O_0 / \dots / O_{v-1} / O$, $v > 0$. Let $Pk \subseteq K_{O_{v-1}}$, where $K_{O_{v-1}}$ is the key of O 's parent O_{v-1} . $K_o = Pk \cup K$. K_o is a *candidate key* of O iff $(Pk, K \rightarrow O)$ and there is no proper subset S of $\{P_k, K\}$ such that $S \rightarrow O$.

We use the following notation for the key of an object type O in a semistructured schema D : $K_o = O_1[\text{@X}_1] / \dots / O_i[\text{@X}_i] / \dots / O_n[\text{@X}_n] / O[\text{@X}]$, where O_i is an object type in D , X_i is a set of O_i 's components, and $O_1 / \dots / O_n$ is a path in D ($n > 0$). If n equals one, then K_o is called an *absolute key*. Otherwise it is called a *relative key*.

Example 3.9 Consider a semistructured schema describing books, which has path $/\text{book}[\text{@isbn}]/\text{chapter}[\text{@number}]/\text{chapter}[\text{@number}]$. We have

- $K_{\text{book}} = \text{book}[\text{@isbn}]$. K_{book} is an absolute key indicating that *isbn* uniquely identifies a book.
- $K_{\text{chapter}} = \text{book}[\text{@isbn}]/\text{chapter}[\text{@number}]$. K_{chapter} is a relative key indicating that a chapter number uniquely identifies a chapter contained in a book.

- $K_{section} = book[@isbn]/chapter[@number]/section[@number]$. $K_{section}$ is a relative key indicating that a section number uniquely identifies a section within a chapter of a book.

[10] defines a foreign key constraint for semistructured data which is similar to IDREF in XML. Consider the schema in Figure 2.2(b). sid in $student_1$ is a foreign key and refers to $student_2$ (referenced object type). This reference semantics is represented graphically as a dashed edge (*reference edge*) from the foreign key to the referenced object type.

Definition 3.9: Let D be a semistructured schema and O be its root object type. The set of *basic dependencies* of D , denoted as $BD(D)$, is defined as follows:

- Let X, Y be children of O , non-trivial extended functional dependencies of the form $X \Rightarrow Y$ where X is a key of O or Y is part of a key of O , are in $BD(D)$.
- For each sub-object type O_i of O , extended functional dependency $K_O \Rightarrow O_i$ is in $BD(D)$, where K_O is O 's key.
- Let O_I be a complex sub-object type of O and D_I be a schema tree that is rooted at O_I and add K_O as attribute(s) of O_I , then $BD(D_I) \subseteq BD(D)$.
- No other non-trivial dependencies that are not generated from above is in $BD(D)$

3.3 NF-SS: Normal Form for Semistructured Schemata

We now give the definition of NF-SS.

Definition 3.10 Let D be a semistructured schema and O be its root object type. D is in *Normal Form for Semistructured Schemata* (NF-SS), iff

1. O has at least one key.
2. For any non-trivial EFD of the form $X \Rightarrow Y$ satisfied by O , where X and Y are attributes or atomic sub-object types of O , then either X is a key or Y is part of the key of O .
3. For any complex sub-object type O_I of O
 - (a) If adding K_O to O_I as its components with other remains, a schema tree D_I rooted at O_I will be in NF-SS.
 - (b) $K_O \cap K_{O_I} = \emptyset$ or $K_O \subset K_{O_I}$, where K_O and K_{O_I} are O and O_I 's key respectively.
 - (c) O_I is not transitively dependent on K_O
4. Any non-trivial EFD in D can be derived from $BD(D)$ by using the inference rules for EFDs.

4. Designing Semistructured Schema in NF-SS

We adopt the restructuring approach to design semistructured schema. We propose a set of restructuring rules to transform a semistructured schema into NF-SS. This restructuring involves the decomposition of object types, creation of new object types

and regrouping of components in a semistructured schema. The objective is to remove transitive or partial EFD and incoherent EFD. This is accomplished by identifying violations of the conditions of NF-SS from the given dependency and key constraints.

4.1 Normalization Rules

Rule 1. (Remove Transitive Dependencies by Decomposition) Given an object type O in a semistructured schema D . Suppose there exists some non-prime component(s) Y of O that is transitively dependent on some key K_O of O , i.e., $K_O \Rightarrow X$, $X \Rightarrow Y$ and $X \not\Rightarrow K_O$, and $X \cap K_O = \emptyset$. Then D can be restructured as follows:

1. Duplicate X to form a new node(s) Z .
2. Move Y and all the descendants of Y and their corresponding edges under Z .
3. Make X as foreign key of O , and add a reference edge from the original node X to Z .

Element duplication is actually relation decomposition or splitting when the element is an object type. Rule 1 can be used to remove undesirable transitive dependency. Note that splitting may happen in many ways and choosing a correct way to decompose is nontrivial since certain splitting will cause the loss of EFDs.

Example 4.1 Consider the schema in Figure 2.2(a) with the following dependency constraints:

- (1) department[@name] $\Rightarrow$ course[@cid] (2) course[@cid] $\rightarrow$ department
 (3) course[@cid] $\rightarrow$ course[@title] (4) course[@cid] $\Rightarrow$ student[@sid]
 (5) course[@cid], student[@sid] $\rightarrow$ grade (6) student[@sid] $\rightarrow$ student[@name, @age]

From the EFDs (4) and (6), D is not in NF-SS since *name* and *age* are transitively dependent on *course* via *student*. Furthermore, $\text{student}[@sid] \not\Rightarrow \text{course}$. Since $\text{course}[@cid] \cap \text{student}[@sid] = \emptyset$, we can use Rule 1 to decompose *student* into two object types: *student1*(sid; grade) and *student2*(sid, name, age). The former remains as a sub-object type of *course* while the latter becomes the root of a new schema tree. A reference edge is created from *student1*'s sid (foreign key) to *student2*. The schema is now in NF-SS, as shown in Figure 2.2(b).

Rule 2. (Remove Path Anomaly by Path Splitting) Given a semistructured schema D . Suppose there exists an incoherent EFD: $O_1[@X_1], \dots, O_n[@X_n] \mapsto Y$, where $\mapsto$ denotes either $\rightarrow$ or $\Rightarrow$, Y is either an object type or an attribute, and there exists a path P that contains $\{O_1, \dots, O_n, Y\}$. Path P can be split into two sub-paths P_1 and P_2 , where P_1 only contains $\{O_1, \dots, O_n\}$ and Y , while P_2 contains $\{O_1, \dots, O_n\}$ and $(P-Y)$. If we have $\mapsto$ is $\rightarrow$, then the cardinality of Y (when Y is an object type) is assumed to be “?” after restructuring; Otherwise, if we have $\mapsto$ is $\Rightarrow$, then the cardinality of Y is “*”.

Rule 2 can be used to remove path anomalies and partial dependencies. This in turn helps to avoid *over-nesting*. Intuitively, components (whether they be attributes or object types) should be kept as close to the *owner* object type as possible. This is achieved though path splitting. The *promotion* of a component may happen more than once, each time moving the component closer to its rightful owner.

Example 4.2 Consider the schema D shown in Figure 3.1(a). Assume the set of specified EFDs follows:

(1) $\text{teacher}[\text{@tid}, \text{time}] \rightarrow \text{ClassRoom}$ (2) $\text{teacher}[\text{@tid}, \text{time}] \rightarrow \text{subject}$

D is not in NF-SS since (1) is an incoherent dependencies. Applying Rule 2, D is normalized by splitting path $/\text{teacher}/ \text{classroom}/\text{subject}/\text{time}$ into two sub-paths: $/\text{teacher}/ \text{time}/\text{ClassRoom}$ and $/\text{teacher}/\text{time}/\text{subject}$, as shown in Figure 3.1(b).

Rule 3. (Removing Partial Dependency by Creating New Object type) Given an object type O in a semistructured schema D . Let X be a set of prime attributes of O , and Y be the set of O 's attributes. Let O_1 be a sub-object type of O . If $(K_O - X) \Rightarrow O_1$ and no proper superset of X satisfy this property, then D can be restructured as follows:

1. $(K_O \cap Y - X)$ becomes the only attribute(s) of O while O_1 remains to be its sub-object type.
2. Create a new object type O_2 that is a direct component of O .
3. Move rest of the components of O and all their descendants and corresponding edges under O_2 .

Example 4.3 Consider the schema D in Figure 4.1(a). Suppose O satisfies the EFDs $\{O[\text{@A}, \text{@B}] \rightarrow D, O[\text{@A}, \text{@B}] \Rightarrow O_2, O[\text{@A}] \Rightarrow O_1, O[\text{@A}] \rightarrow E\}$ and the key of O is $\{A, B\}$.

D is not in NF-SS because $O[\text{@A}, \text{@B}] \Rightarrow O_1$ is not a full dependency. Applying Rule 3, a new object O_3 is created which has O_2 and $\{B, D, E\}$ as its children (O is renamed as O'). The restructured schema given in Figure 4.1(b) is still not in NF-SS since there exists an incoherent dependency $O'[\text{@A}] \rightarrow E$. We apply Rule 2 to promote E as a child of O' (O' is renamed as O''). The schema obtained in Figure 4.1(c) is now in NF-SS.

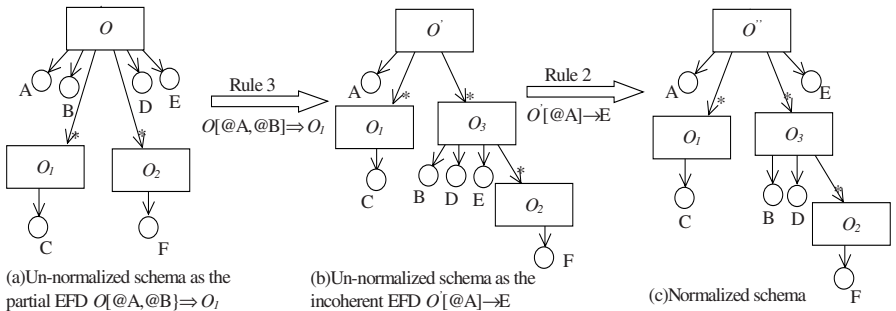


Fig. 4.1: Using Rules 2 and 3 to restructure a schema.

Rule 4. (Restructure to satisfy Condition 3(b) of NF-SS Definition) Given an object type O in a semistructured schema D . Let X be a set of O 's attributes and single-valued atomic sub-object types, O_1 be a complex sub-object type of O . O_1 has

relative key K_{O_1} , but $K_O \not\subset K_{O_1}$ and $K_{O_1} \not\rightarrow K_O$. Let Y be $K_O \cap K_{O_1} \cap X$, and $Y \neq \emptyset$. D can be restructured as follows:

1. O_1 remains to be a sub-object type of O .
2. Make Y as components of O .
3. Create a new object type O_2 to be a child of O and the rest components of O (excluding Y) become children of O_2 .

Example 4.4 Consider the semistructured schema D in Figure 4.2(a). Suppose O satisfies the EFD: (1) $O[\text{@K}, \text{@A}] \Rightarrow O_1$ (2) $O[\text{@K}, \text{@B}] \Rightarrow O_2$ and the key of O is $\{K, A, B\}$.

D is not in NF-SS since O_1 and O_2 are partially dependent on the key of O . We use Rule 3 to create a new object type O_3 , rename O as O' and make it a child of O' ; After that, we move B and O_2 and all their descendants and corresponding edges under O_3 . Figure 4.2(b) shows the schema obtained. This is still not in NF-SS because Condition 3(b) in the NF-SS definition remains violated: $K_{O'} = O'[\text{@K}, \text{@A}]$, while $K_{O_3} = O'[\text{@K}]/O_3[\text{@B}]$. In addition, $K_{O_3} \rightarrow K_{O'}$ cannot be derived. Applying Rule 4, O_3 remains to be a sub-object type of O' , and K become attribute of O' (O' is renamed as O''). We create a new object type O_4 as a child of O'' and move the rest components of O' and their corresponding edges under O_4 . The schema shown in Figure 4.2(c) is now in NF-SS.

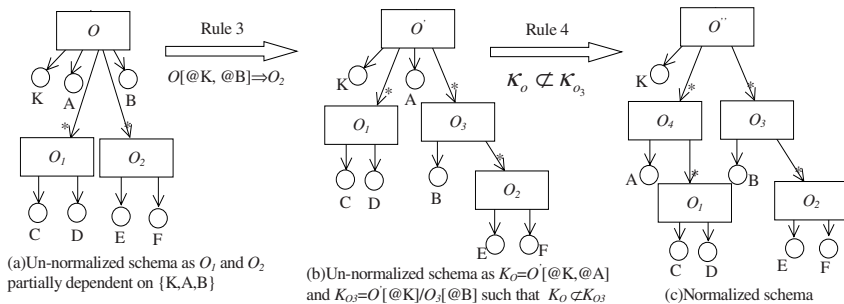


Fig. 4.2: Using Rules 3 and 4 to restructure a schema.

4.2 Restructuring Algorithm

In this section, we present an algorithm that uses the normalization rules presented in Section 4.1 to iteratively restructure a semistructured schema into NF-SS. The algorithm takes as input a set of semistructured schema and a set of dependency constraints for these schemas. It returns as output a set of semistructured schema in NF-SS.

Algorithm 4.1: Restructuring Algorithm

Input: A set S that contains semistructured schemas, and a set of EFDs for S .

Output: A set of semistructured schemas that in NF-SS.

Begin

1. for each semistructured schema D in S do
 - if D is not in NF-SS then repeat until no further change:
 - (1) if $\exists$ transitive EFD: $K_O \Rightarrow X, X \Rightarrow Y$ and $X \not\Rightarrow K_O$ for an object type O in D ,
 - Case 1: $X \cap K_O = \emptyset$.
Apply Rule 1 to remove the transitive EFD.
 - Case 2: $X \subset K_O$.
Apply Rule 3 to remove the transitive EFD.
 - Case 3: $X \cap K_O \neq \emptyset$.
Apply Rule 4 to remove the transitive EFD.
 - (2) if there exists incoherent EFD then apply Rule 2 to remove it.
2. output S .

End

In the normalization process, object types may be created and components of the schema may be regrouped. Two problems are involved there. One is the cardinality of a new sub-object type. We assume “*” on the new object type at this design stage, and let designers/users elaborate later. The other is naming of the new object types. We believe it is generally preferable to have designers/users specify alternate names, which indicate the role played by the object type in the context of the application.

4.3 Discussion

We have presented a technique for restructuring semistructured schema to obtain NF-SS. We would like to highlight two pertinent issues for this restructuring approach. The first issue is the completeness of the restructuring rules. That is, given a schema, is it always possible to restructure it into a set of semistructured schema in NF-SS using heuristic rules such as Rules 1 to 4? This is a difficult question that also arises when the decomposition approach is used in relational databases. It is not always possible to get all the EFDs satisfied by a semistructured schema, that is, covering is not guaranteed. Furthermore, it is not always possible to preserve dependencies during transformation, that is, dependency preservation is not guaranteed, which is a problem that also happens to the decomposition method taken in relational database design. A formal investigation of the problem is beyond the scope of this paper. Nevertheless, we would like to point out that losing some EFD could actually prevent infinite loop for Algorithm 4.1 in some situations.

Consider the following two EFD (1) $A, B \Rightarrow C$ (2) $A, C \Rightarrow D$ for a schema D that has a path $/A/B/C/D$. There is a “conflict” between (1) and (2) in the sense that only one of them is expressible in D . Hence, applying these rules to an unnormalized schema results in infinite schema transformations and there may exist conflicts among the specified EFD constraints.

The second issue is the uniqueness of the solution. That is, does the process of restructuring give a unique solution? The answer is no. In the normalization of

relational schema, it is well known that decomposition does not guarantee unique results as it depends on the order in which the dependencies are examined. Although the restructuring approach does not necessarily give unique results and guarantee dependency preservation, it does give practical heuristics and provides insights into the normalization task for semistructured databases.

5. Related Works

To the best of our knowledge, only [6] and [5] provide works that parallel our research efforts here. [6] defines a schema called S3-Graph. S3-Graph makes no distinction between element node and attribute node and does not specify cardinality on the schema. Therefore S3-Graph doesn't lend itself to XML definition. To identify redundancy in S3-Graph, [6] defines a dependency constraint called SS-Dependency. A S3-Graph is in S3-NF if there is no transitive SS-dependency. This limits the types of redundancies that can be resolved by S3-NF. S3-NF deals with SS-dependency constraints and does not handle key constraints, an essential feature in database design. Furthermore, S3-NF may not remove anomalies such as like partial dependency and path anomaly. In contrast, NF-SS is designed to handle more general situations and therefore, subsumes and extends that of [6].

[5] defines a normal form called XNF (XML Normal Form) is defined. The work in [5] focuses on how to translate a schema, that is represented in conceptual-model hypergraphs, to a scheme-tree forest in XNF. A scheme-tree forest F is in XNF if each scheme tree in F has no potential redundancy with respect to a specified set of (functional and multivalued) constraints C and F has as few, or fewer, scheme trees as any other schemes-tree forest corresponding to M in which each scheme tree has no potential redundancy with respect to C . CM hypergraph has no hierarchical structures, no key concepts; additionally, it has no concept of attributes resulting too many objects in a schema. Although an XNF-compliant DTD can ensure complying XML documents have as few hierarchies as possible, the presented algorithms for generating XNF scheme-tree forest suffers from efficiency. A large set of scheme-tree forests that in XNF is generated and this requires the user to select the best that satisfies their application requirements.

[7] studies the normalization for nested relational data model, and proposes a normal form called NF-NR(*Normal Form for Nested Relations*). Our NF-SS definition and normalization process is similar to that of [7] in concept, but is different in essence, which we have mentioned in the first section.

6. Conclusion

In this paper, we have shown the importance of designing good semi-structured databases. We defined a semistructured schema for semistructured databases, and incorporated it with integrity constraints such as dependency and key constraints. We

identified various anomalies, including rewriting anomaly, insertion anomaly, deletion anomaly and path anomaly, that may arise when a semistructured database is not designed properly and contains redundancies. We proposed NF-SS, a normal form for semistructured schema. A semistructured schema in NF-SS does not have redundancy and hence no undesirable updating anomalies for the conforming semistructured databases. In addition, a semistructured schema in NF-SS also gives a more reasonable representation of real world semantics. We have presented a set of heuristic restructuring rules and developed an algorithm for iteratively restructuring a semistructured schema into NF-SS.

Future directions for research include investigating additional heuristic restructuring rules as well as extending existing rules to deal with various anomalies that may exist in semistructured schemata. We also intend to improve the restructuring Algorithm 4.1 by developing a conflict-detecting framework to check for the existence of conflicts within the specified dependency constraints for a schema.

References

1. S. Abiteboul, R. Hull and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995
2. T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. 2nd Edition, Oct. 2000. <http://www.w3.org/TR/REC-xml>.
3. P. Buneman, S. Davidson, W. Fan, C. Hara and W. Tan. Keys for XML. *Proceedings of the 10th International World Wide Web Conference*, 2001.
4. J. Clark and S. DeRose. XML Path Language (XPath). W3C Working Draft, November 1999. <http://www.w3.org/TR/xpath>.
5. D.W.Embley and W.Y.Mok. Developing XML Documents with Guaranteed “Good” Properties. *Proceedings of the 20th International Conference on Conceptual Modeling (ER)*, 2001.
6. S. Y. Lee, M. L. Lee, T. W. Ling and L. A.. Kalinichenko. Designing Good Semistructured Databases. *Proceedings of the 18th International Conference on Conceptual Modeling (ER)*, 1999.
7. T. W. Ling and L. L. Yan. NF-NR: A Practical Normal Form for Nested Relations. *Journal of Systems Integration*. Vol4, 1994, pp309-340
8. Z. M. Ozsoyoglu and L. Y. Yuan. A New Normal Form for Nested Relations. *ACM Transaction on Database Systems*. 12(1), (1987).
9. R. Ramakrishnan and J.Gehrke. *Database Management Systems*. McGraw-Hill Higher Education, 2000.
10. Xiaoying Wu. Designing Good Semistructured Databases. Master Thesis, School of Computing, National University of Singapore, 2002.

A Formal Analysis of the Lightweight Directory Access Protocol

Fang Wei and Georg Lausen

University of Freiburg, Institute for Computer Science, Germany
`{fwei,lausen}@informatik.uni-freiburg.de`

Abstract. LDAP (Lightweight Directory Access Protocol) directories are being widely used on the Web, for white pages information, user profiles, etc. The advantages LDAP offers are (i) the support for highly distributed data on the Web while still keeping a uniform data model; (ii) the flexibility of a semi-structured data model, i.e. a flexible data type definition enabling the presentation and manipulation of heterogeneous data entries in a natural manner. Although many implementations of the LDAP protocol exist, the still lacking logical formalization prohibits a formal analysis and makes it difficult to make use of the numerous results developed for relational databases. In this paper, we give a first-order logic semantics of LDAP and discuss the expressive power of LDAP. In particular, schema typing constraints are interpreted as semantic integrity constraints. We apply our framework to the containment problem of LDAP queries with schema constraints; we reduce this problem to the containment problem of Datalog in the presence of integrity constraints.

1 Introduction

LDAP (Lightweight Directory Access Protocol) directories are being widely used on the Web. Besides the traditional directory application like white pages information, user profiles, etc, it has been deployed in a number of commercial directory server implementations [12] (also see [11] for a survey). The Internet Engineering Task Force (IETF) has standardized the Lightweight Directory Access Protocol (LDAPv3) for modeling and querying network directory information [22].

An LDAP-based network directory can be viewed as a highly distributed database, in which the directory entries are organized in a hierarchical namespace and can be accessed using database-style search functions. The advantages LDAP offers are (i) the support for highly distributed data on the Web while still keeping a uniform data model; (ii) the flexibility of a semi-structured data model, i.e. a flexible data type definition enabling the presentation and manipulation of heterogeneous data entries in a natural manner. Due to the similarities of the data model of LDAP and XML, there have been works on the transformation from XML to LDAP [15].

In spite of the various implementations of the LDAP protocol, the still lacking logical formalization prohibits a formal analysis and makes it difficult to make use of the numerous results developed for relational databases. In this paper, we seek to analyze the first order semantics of a full-fledged LDAP query language and make the following contribution:

- We give a formal description of the LDAP directory data model in Section 2, analyze the expressive power of the LDAP query language, and present a complete algorithm to transform LDAP queries to *Datalog*.
- Query optimization is of general interest in traditional databases. Since determining equivalence of queries is one of the most fundamental query optimization problems, we demonstrate the query equivalence or query containment problem of LDAP in the first order logic semantics (Section 2.3).
- Integrity constraints are crucial in semantic query optimization and view updating. In Section 3 we transform the LDAP schema information into a form of generalized integrity constraints and tackle the containment problem in the presence of such integrity constraints.

Section 4 compares our results to previous work in the literature, and finally Section 5 concludes this paper.

2 LDAP Directory Model

The formal description of LDAP directory can be found in [4,7] and the informal one in [10]. However, the form and syntax of these descriptions make them extremely difficult to be directly applicable to the techniques and results found in the literature. Therefore, we need to abstract the peculiarities of the LDAP syntax (Section 2.1), to then reconstruct the model in standard logical terminology (Section 2.2).

2.1 General Model

An LDAP directory consists of the following two main components:

Directory Schema: Defines a finite set of classes, attributes, and types. Each attribute has exactly one type, and each class specifies a set of **required** and **allowed** attributes.

Directory Instance: Contains a finite set of entries organized in a forest, where each entry

- belongs to at least one class,
- has a non-empty set of (possibly) multivalued attribute-value pairs that conform to the schema definition,
- defines at least attributes *oc* and *dn*, where *oc* determines what classes the entry belongs to, and *dn* provides a unique distinguished name for the entry,
- is placed in the instance hierarchy based on its distinguished name.

In order to show an example of LDAP schema and instance, we take directly from [7]. All the examples refer to the schema given in Table 1, which shows how an LDAP schema is defined. Figure 1 gives an LDAP instance; note each entry has a unique *distinguished name* (DN) which acts as a path expression in the semi-structured data model. The directory describes the whole organization, contains sub-directories for e.g. persons, groups, etc. The bottom entry describes a person, and it is the instance of the class `inetOrgPerson` and `external` which is defined in Table 1 (Here the attribute `oc` means `ObjectClass`). Furthermore, it has a set of (attribute, value) pairs. Note that multi-value attributes are allowed here, e. g. the entry has two values for the attribute `sn`, `cluet` and `verduron`. In

Table 1. Schema of an example LDAP directory

class	external	inetOrgPerson	internal	residentialPerson
requires	oc, mail, uniqueness	oc,sn,cn	oc,uniquelid,cn	oc,sn,cn,locality
allows	cn	tel,mail,age	mail,sn,gender,locality,tel	tel

addition, LDAP offers a rather limited query facility based on filter definitions. A query is defined by a filter, which consists of the following four components:

Base: The *distinguished name* (DN) of the entry in the directory instance where the search will start. The DN is a sequence $[s_1, \dots, s_n]$, where each of the s_i is an attribute-value pair (attr, val). The first pair, s_1 , is called *relative distinguished name* (RDN), which distinguishes the entry from its sibling entries. For example, the DN of the bottom entry of Figure 1 is: $[(\text{name}, \text{cluet}), (\text{ou}, \text{people}), (\text{o}, \text{att.com})]$, its RDN is (name, cluet).

Scope: Can be `base`¹, if the search is to be restricted to just the first node, `one`, if only the first level of nodes is to be searched, or `sub`, if all nodes under the base should be considered by the filter expression.

Filter Expression: Defined as the boolean combination of atomic filters of the form $(a \text{ op } t)$, where:

- a is an attribute name;
- op is a comparison operator out of the set $\{=, <, \leq, >, \geq\}$;
- t is an attribute value, or $*$, used to test for existence of an attribute.

The boolean operators are **and** (&), **or** (|) and **not** (!).

Projection: Defines a set of attribute names that determines what data should be returned from each matching entry.

Example 1. Consider the query defined by the LDAP filter

$$([(ou, people), (o, att.com)], sub, \{\&(sn = olga*)(age < 25)\}, \{oc, sn\})$$

¹ It is different from the **Base** above. We write from now on **Base** for the root entry and **base** for the scope.

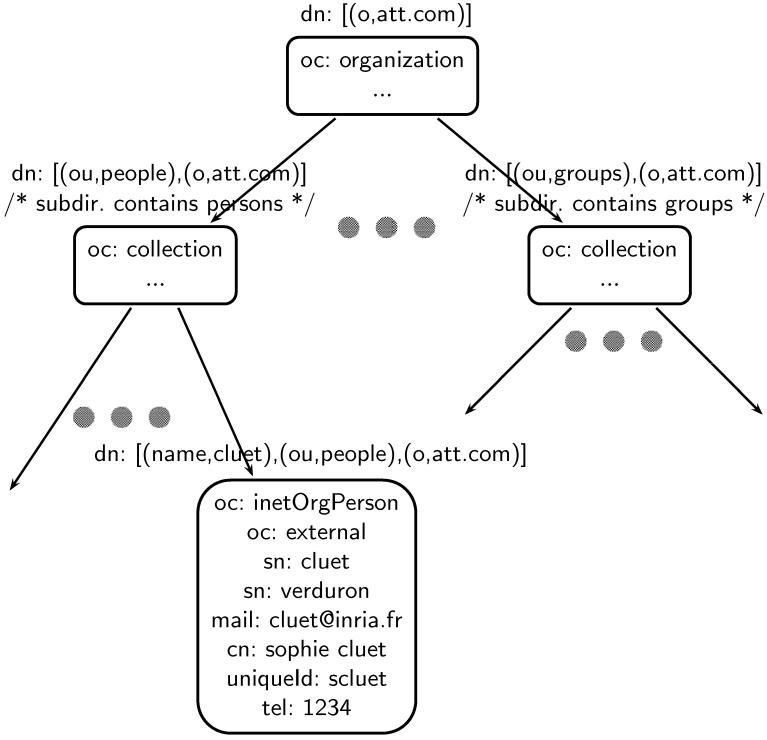


Fig. 1. LDAP instance example

would start the search at the $[(ou,people), (o,att.com)]$ entry, test all its descendants against the filter expression, and return only the *oc* and *sn* attribute-value pairs.

2.2 Logical Framework

In order to build our logical framework, we first need to formalize the elements that make up the definition of the directory schema and the directory instance as far as they are relevant for our purposes. Based on these definitions we give a logical reformulation of an LDAP directory schema and instance. Later we shall introduce a set of additional axioms that capture the nature of an LDAP directory more comprehensively.

Definition 1 (Directory Schema). *A directory schema S is a 4-tuple $S = (\mathcal{C}, \mathcal{A}, \mathcal{D}, \rho)$, such that:*

- $\mathcal{C}$ is a finite set of classes.
- $\mathcal{A}$ is a finite set of attributes such that $\{oc, dn\} \subseteq \mathcal{A}$.

- $\mathcal{D}$ (Domain) is an infinite ordered set containing all possible values an LDAP attribute can take. In particular, this set contains all class names, attribute names, and distinguished names.
- ρ is a pair of functions $\rho_1, \rho_2 : \mathcal{C} \rightarrow 2^{\mathcal{A}}$ that specify what attributes are needed for each class. ρ_1 defines the required attributes and ρ_2 the allowed attributes. Moreover, $\rho_1(c) \cap \rho_2(c) = \emptyset$ and $\{oc, dn\} \subseteq \rho_1(c)$, for all $c \in \mathcal{C}$. $\square$

Definition 2 (Directory Instance). A directory instance I is a 4-tuple $I = (\mathcal{E}, \gamma, \delta, \alpha)$, such that:

- $\mathcal{E}$ is a finite set of directory entries.
- γ is a function of the form $\gamma : \mathcal{E} \rightarrow 2^{\mathcal{C}}$ that associates each entry with a non-empty set of classes from $\mathcal{C}$.
- δ assigns a unique distinguished name to each entry in $\mathcal{E}$.
- α is a function of the form $\alpha : \mathcal{E} \rightarrow 2^{\mathcal{A}' \times \mathcal{D}}$, $\mathcal{A}' = \mathcal{A} \setminus \{oc, dn\}$, that associates a set of attribute-value pairs to each entry. $\square$

Example 2. Consider the entry in Figure 1, we have:

$$\alpha([(ou, people), (o, att.com)]) = \{(sn, cluet), (sn, verduron), (mail, cluet@inria.fr), (cn, sophie cluet), (uniqueid, scluet), (tel, 1234)\}$$

A logical reformulation of an LDAP directory schema and instance then can be stated as follows:

Definition 3 (LDAP Relations). Let $S = (\mathcal{C}, \mathcal{A}, \mathcal{D}, \rho)$ be an LDAP schema and $I = (\mathcal{E}, \gamma, \delta, \alpha)$ a LDAP instance to S .

- An LDAP schema S is represented by a relation $schema(class, attr, req)$, where $class \in \mathcal{C}$, $attr \in \mathcal{A}$, and req is a boolean value that indicates whether an attribute $attr$ is required or allowed by $class$. An instance of $schema$ is defined according to ρ : $schema(c, a, \mathbf{true})$ iff $a \in \rho_1(c)$, respectively, $schema(c, a, \mathbf{false})$ iff $a \in \rho_2(c)$, where $c \in \mathcal{C}$ and $a \in \mathcal{A}$.
- An LDAP instance I is represented by a relation $inst(entry, attr, val)$, where $entry \in \mathcal{E}$, $attr \in \mathcal{A}$, and $val \in \mathcal{D}$. Instances of $inst$ are defined according to γ , δ and α . $(e, oc, v) \in inst$ iff $v \in \gamma(e)$, $(e, dn, v) \in inst$ iff $v = \delta(e)$, and $(e, a, v) \in inst$ iff $(a, v) \in \alpha(e)$, where $e \in \mathcal{E}$, $a \in \mathcal{A}$, and $v \in \mathcal{D}$. $\square$

The unique schema information in LDAP makes the query processing and optimization a more difficult task than for conjunctive queries over a relational schema: there are several *hidden* rules implied by the schema definition which do not exist in the traditional relational model (e.g. the attribute of a class can be either **required**, or **allowed**). In Section 3 a set of integrity constraints originated from LDAP schema will be logically formalized and play an important role in semantic query optimization. **Multivalued attributes.** As introduced

above, when an entry has multi-valued attributes, then the transformation of the operator '!' is not trivial any more. Consider the example entry in Figure 1 where the attribute `sn` has two values `cluet` and `verduron`. To illustrate, let the filter be $!(sn = cluet)$, then it is semantically not equal to the transformation $inst(X, sn, Z), Z \neq cluet$. The first one will yield “false”, whereas the second one returns the answer $X = [(name, cluet), (ou, people), (o, att.com)], Z = verduron$. Instead, we have to cope with the following cases: (i) the entry does not have `sn` as attribute, that is, the attribute `sn` is not defined; (ii) even if the entry has `sn` defined as attribute, but it has two values for it (like in our example), then the transformation will yield false results too. As a result, we need to define the additional predicates shown in Definition 4 and 5.

Taking into account that attributes may or may not be defined for certain entries, we are now going to introduce two further relations `def` and `undef` as follows:

Definition 4 (Definedness). *Let $def(entry, attr)$ and $undef(entry, attr)$ be relation schemata, where $entry$ is of type $\mathcal{E}$ and $attr$ is of type $\mathcal{A}$. Instances to def and $undef$ are given by:*

- $def(X, Y)$ iff $\exists Z : inst(X, Y, Z)$.
- $undef(X, Y)$ iff $\nexists Z : inst(X, Y, Z)$.

In both cases the domain of quantification of the variables X, Y, Z is $\mathcal{E}, \mathcal{A}$, respectively, $\mathcal{D}$. $\square$

Note, that according to this definition `def` and `undef` are guaranteed to be finite. We can understand `undef` as a *safe* negation of `def`, i.e., $\forall X \in \mathcal{E}, \forall Y \in \mathcal{A} : (\neg def(X, Y) \equiv undef(X, Y))$. In order to give the negation operator '!' the correct semantics, we have to first introduce a set of predicates:

Definition 5 (Arithmetics). *Let $eq(entry, attr, val)$, $greater(entry, attr, val)$, $less(entry, attr, val)$, and $greater\_eq(entry, attr, val)$, $less\_eq(entry, attr, val)$ be relations, where $entry \in \mathcal{E}$, $attr \in \mathcal{A}$, and $val \in \mathcal{D}$. Instances of the relations are given by:*

$$\begin{array}{ll}
 eq(X, Y, V) & :- inst(X, Y, Z), Z = V. \\
 greater(X, Y, V) & :- inst(X, Y, Z), Z > V. \\
 less(X, Y, V) & :- inst(X, Y, Z), Z < V. \\
 greater\_eq(X, Y, V) & :- inst(X, Y, Z), Z \geq V. \\
 less\_eq(X, Y, V) & :- inst(X, Y, Z), Z \leq V.
 \end{array}$$

(See also Example 5). $\square$

Formulas in our framework are built over the relations `inst`, `schema`, `def`, `undef`, `eq`, `greater`, `less`, `greater_eq`, `less_eq` and arithmetic predicates as used in the LDAP filter language. It will suffice to restrict our logical language to formulas in DNF (Disjunctive Normal Form), as we shall see soon. Moreover, because of the filter syntax, all arithmetic predicates have the *semi-interval* property [13].

In the sequel, we denote an LDAP filter by F^* and its filter expression by F . Any LDAP filter expression F can be transformed into the logical model as follows:

In the following, X denotes a distinguished variable reused in any transformation step; in contrast, variable Z in any transformation step will represent a new variable, say $Z_1, Z_2, \dots$, distinct from all variables previously introduced.

1. Replace '&' by ' $\wedge$ ', '|' by ' $\vee$ ', and '!' by ' $\neg$ ', and transform the resulting formula into DNF.
2. Transform every positive occurrence of $(a = *)$ into $\text{inst}(X, a, Z)$, every negative occurrence $!(a = *)$ to $\text{undef}(X, a)$.
3. Consider now expressions different to $(a = *)$. Transform every positive occurrence of $(a \text{ op } v)$ into the conjunction $(\text{inst}(X, a, Z) \wedge (Z \text{ op } v))$, and every negative occurrence $!(a \text{ op } v)$ into the formula $\neg f(\text{op})(X, a, v)$ ².
4. Reestablish disjunctive normal form and call the resulting expression $\Sigma(F)$.

Example 3. Following the transformation algorithm, the filter

$\&(\text{uniqueId} = \text{olga*})(!(\text{mail} = *))$ is transformed to

$\text{inst}(X, \text{uniqueId}, Z) \wedge (Z = \text{olga*}) \wedge \text{undef}(X, \text{mail})$. In contrast, the filter

$\&(\text{uniqueId} = \text{olga*})(!(\text{mail} = \text{olga*}))$ is transformed to

$\text{inst}(X, \text{uniqueId}, Z_1) \wedge (Z_1 = \text{olga*}) \wedge \neg \text{eq}(X, \text{mail}, \text{olga*})$. □

Next the **Scope** and **Base** parts of the LDAP query are considered. We borrow the list expression syntax of Prolog to represent the DN, such that each DN of any entry is a list, in which each element is an attribute-value pair. As a result, we introduce two new predicates **one** and **sub** which have the meaning of **child** and **descendant** in XPath respectively.

Definition 6 (LDAP Scope Relation). *The semantics of $\text{one}(\text{dn1}, \text{dn2})$ and $\text{sub}(\text{dn1}, \text{dn2})$ are logically described as follows: where $\text{dn1} \in \mathcal{E}$ and $\text{dn2} \in \mathcal{E}$:*

- $\text{one}(\text{Parent}, [\text{RDN}|\text{Parent}])$. (*fact*)
- $\text{one}(X, Y) \rightarrow \text{sub}(X, Y)$. (*base*)
- $\text{one}(X, Y), \text{sub}(Y, Z) \rightarrow \text{sub}(X, Z)$. (*trans*)

Together with *fact*, the rule *trans* can be rewritten to the following constraint which means that if Z is the descendant of one node, then it is also the descendant of the parent of that node:

- $\text{sub}([\text{RDN}|\text{Parent}], Z) \rightarrow \text{sub}(\text{Parent}, Z)$. (*trans'*)

² where $f(\text{op})$ is the function of the five operators as the following: $f(=) = \text{eq}$, $f(>) = \text{greater}$, $f(<) = \text{less}$, $f(\geq) = \text{greater\_eq}$, and $f(\leq) = \text{less\_eq}$.

Note that the rules **fact** and **trans**' have the form of *inclusion dependencies*. In [25], a generalized form of *inclusion dependencies* called *referential constraints* is introduced and the query containment problem in the presence of such constraints is studied. In the later part of the paper, we will give an example to show how the *referential constraints* affect the LDAP query containment checking.

The transformation of the **Base** and **Scope** is simple. The additional predicate $\text{sub}(\mathbf{Base}, X)$ for the scope **sub** and $\text{one}(\mathbf{Base}, X)$ for **one** is added to the filter expression F respectively. Note that the variable X is the same one as in the filter expression, and **Base** is always the DN of the given root entry. As for the scope **base**, one arithmetic predicate ($X = \mathbf{Base}$) is needed to be added in F , where X must also be the same variable as that in F . **The complete transformation.** $\Sigma(F)$ is the disjunction of the form $\{G_1, \dots, G_k\} (k \geq 1)$ and each $G_i (1 \leq i \leq k)$ has the form:

$$R_1(\bar{Y}_1) \wedge \dots \wedge R_n(\bar{Y}_n) \wedge \neg P_1(\bar{W}_1) \wedge \dots \wedge \neg P_m(\bar{W}_m) \wedge C_Q$$

in which $R_1, \dots, R_n (n \geq 1)$ are taken from the relations $\{\text{inst}, \text{def}, \text{undef}, \text{sub}, \text{one}\}$; $P_1, \dots, P_m (m \geq 0)$, are predicates as given in Definition 5; and C_Q is composed of predicate expressions $u\theta v$, where $\theta \in \{=, \neq, <, >, \leq, \geq\}$, u is the element of $\bar{Y}_1, \dots, \bar{Y}_n$, and v is a constant. Furthermore, for the sake of *safe negation*, we have $\{\bar{W}_1, \dots, \bar{W}_m\} \subseteq \{\bar{Y}_1, \dots, \bar{Y}_n\}$, to ensure that all the variables in negated predicates appear in the positive ones.

Now the projection part of an LDAP filter has to be taken into account. In the logical model, any projected attribute gives rise to a conjunctive query. In particular, because $\Sigma(F)$ can be considered a set of conjunctions, for each projected attribute a we get a set $Q_a = \{Q_{a1}, \dots, Q_{ak}\}$ of conjunctive queries, each $Q_{ai}, 1 \leq i \leq k$, has the form

$$q_{ai}(X, a, Z) :- \text{inst}(X, a, Z) \wedge G_i.$$

For each conjunctive query the variable Z is distinct from X and from all variables in the respective G_i , and is used to *copy* the value of the projected attribute into the query result.³ The domain of quantification of variable X is the set of entries $\mathcal{E}$, where for the Y 's and Z the domain is the set of possible values. To simplify notation, quantifiers are omitted and ' $\wedge$ ' is replaced by comma. The set of conjunctive queries corresponding to an LDAP filter F^* is denoted by $\Sigma(F^*)$. Up to now, we have completely transformed LDAP into the logical formalization. The final transformation is demonstrated by the following example:

Example 4. Again considering the Example 1, the LDAP query is transformed into two conjunctive queries since two attributes **oc** and **sn** should be projected. We have then two queries with **oc** and **sn** at the head respectively.

$$\begin{aligned} q1(X, \text{oc}, Z1) &:- \text{sub}([(ou, \text{people}), (o, \text{att.com})], X), \text{inst}(X, \text{oc}, Z1), \text{inst}(X, \text{sn}, Z2), \\ &\quad \text{inst}(X, \text{age}, Z3), Z2 = olga*, Z3 < 25. \\ q2(X, \text{sn}, Z2) &:- \text{sub}([(ou, \text{people}), (o, \text{att.com})], X), \text{inst}(X, \text{sn}, Z2), \\ &\quad \text{inst}(X, \text{age}, Z3), Z2 = olga*, Z3 < 25. \end{aligned}$$

³ If the literal $\text{inst}(X, a, Z)$ is already contained in G_i , it is omitted.

Theorem 1. *Without considering the LDAP schema, the LDAP query can be equivalently transformed to Datalog with arithmetic comparison and stratified negation.* $\square$

2.3 Containment of LDAP Queries

As long as the transformation of LDAP queries to *Datalog* is complete, we can conclude that the containment problem of LDAP can be reduced to an equivalent containment problem of *Datalog*, which is unfortunately undecidable [19]. However, the only recursions which cause the undecidability are the transitive closure rules, namely **base** and **trans'** in Definition 6. We argue that since we regard the definitions by *referential constraints* on the **one** and **sub** relations, instead of rules which *define* these IDB relations, the transformed query is *non-recursive Datalog* together with *referential constraints* for which query containment is decidable.

Theorem 2. *Containment of basic LDAP queries is decidable, and it is polynomially equivalent to the containment problem of non-recursive Datalog.* $\square$

The general query containment problem in *Datalog* has been extensively studied [19,6]. However, several interesting cases originating from the LDAP query model are worth of discussing here. **Containment of queries with stratified negation.** The containment of Datalog with stratified negation is studied in [14], however, the algorithm provided in [14] tests *uniform equivalence* of *Datalog* queries and actually, the algorithm would fail to prove in the above example that $Q_1 \subseteq Q_2$, which is obviously true. In [24], an algorithm is given to check the containment for (i) conjunctive queries with safe negation, and (ii) non-recursive Datalog with stratified negation which is the case here.

Example 5. Let the filters of two LDAP queries be the following:

$$\begin{aligned} f_1 &: \&((sn = *) !(tel = *)), \\ f_2 &: \&((sn = *) !(tel = 1234*)). \end{aligned}$$

Let both queries have **sn** as projected attribute. Q_1 and Q_2 can then be built up as the following (to simplify the queries, we assume that the **Scope** and **Base** of the queries are identical, so that they are omitted here):

$$\begin{aligned} Q_1 &: q(X, sn, Z) : - \text{inst}(X, sn, Z), \text{undef}(X, tel). \\ Q_2 &: q(X, sn, Z) : - \text{inst}(X, sn, Z), \neg \text{eq}(X, tel, 1234*). \end{aligned}$$

Note that according to Definition 5, $\text{eq}(X, tel, 1234*)$ holds if X has *any* attribute value for **tel** which matches 1234*. $\square$

Containment of queries with scope constraints. As we noticed in the LDAP scope definition, the *referential constraints* are introduced to represent the transitive closure of the **sub** relation (see the rules **base** and **trans'** in Definition 6). As a result, the containment problem in the presence of *integrity constraints* must be considered. In [25], an algorithm for checking conjunctive

query containment together with *implication constraints* and *referential constraints* is given. Due to the size limit of the paper, we only state it informally as the following: if the containment of queries Q_1 and Q_2 ($Q_1 \subseteq Q_2$) is to be checked, we first *expand* the query Q_1 by *referential expansion* using the *referential constraints* according to [25], and then the normal containment checking is executed. The following example shows how the algorithm is applied in the LDAP logic model.

Example 6. Consider the following LDAP queries LQ_1 and LQ_2 :

$$\begin{aligned} LQ_1 &: ([ou=people,o=att.com], one, \{(sn = *)\}, \{sn\}) \\ LQ_2 &: ([o=att.com], sub, \{(sn = *)\}, \{sn\}) \end{aligned}$$

The queries are rewritten as the following Q_1 and Q_2

$$\begin{aligned} Q_1 &: q(X, sn, Z) :- one([(ou,people),(o,att.com)], X), inst(X, sn, Z). \\ Q_2 &: q(X, sn, Z) :- sub([(o,att.com)], X), inst(X, sn, Z). \end{aligned}$$

Query Q_1 asks for entries with attribute sn and are children of the starting entry $[(ou,people),(o,att.com)]$; while Q_2 asks for entries which are descendants of the starting entry $[(o,att.com)]$. From the transitive closure it is easy to see that the answer set of Q_1 is contained in that of Q_2 since the scope of Q_1 is a *subtree* of that of Q_2 .

Formally described, we first expand the atom $one([(ou,people),(o,att.com)], X)$ in the body of Q_1 . Due to the referential constraints **base** and **trans'** above, the expanded conjunction:

$$\{one([ou,people],[o,att.com]), X), sub([ou,people],[o,att.com]), X), sub([(o,att.com)], X)\}.$$

is semantically equivalent to the original atom above. Then, it is not hard to obtain that $Q_1 \subseteq Q_2$ – in the presence of the *referential constraints*. $\square$

3 Schema Integrity Constraints in LDAP

The LDAP data model can represent heterogeneity of entities in a very natural and flexible manner, however, the gain of such flexibility does not come for free. In order to capture the semantics of the LDAP schema, we need to formalize them as integrity constraints, namely *disjunctive referential constraints* and *implication constraints* which will be introduced below. Integrity constraints in traditional databases are used for semantic query optimization, cooperative query answering, and view updating, etc [8]. In the rest of the section, we first describe the *disjunctive referential constraints* and *implication constraints* in the case of LDAP model, then the algorithm of containment checking in the presence of such integrity constraints is given.

Definition 7 (Constraints in LDAP).

Disjunctive Referential Constraint 1: *If an entry has values for attribute $Attr$, then it must belong to at least one class for which $Attr$ is an attribute:*

$$inst(X, Attr, Val) \rightarrow \bigvee_{c \in C, \text{schema}(c, Attr, -)} inst(X, oc, c).$$

Disjunctive Referential Constraint 2: *Each entry must belong to at least one class:*

$$undef(X, Attr) \rightarrow \bigvee_{c \in C} inst(X, oc, c).$$

Implication Constraint: *An entry must have at least one value for each of the required attributes of its classes:*

$$inst(X, oc, C) \wedge \text{schema}(C, Attr, \text{true}) \wedge undef(X, Attr) \rightarrow .$$

Note that *disjunctive referential constraints* are a generalized form of *referential constraints* in [25], *integrity constraints* formalized in [8], and *embedded dependencies* in [3]. The general form of disjunctive referential constraints (DRC) is as follows:

$$\forall (\bar{Y}_1, \dots, \bar{Y}_m) [r_1(\bar{Y}_1), \dots, r_m(\bar{Y}_m) \rightarrow \exists (Z_1, \dots, Z_k) p_1(\bar{X}_1) \vee \dots \vee p_n(\bar{X}_n)].$$

where $p_1, \dots, p_n (1 \leq n)$, and $r_1, \dots, r_m (0 \leq m)$ are predicate names; $\bar{X}_1, \dots, \bar{X}_n, \bar{Y}_1, \dots, \bar{Y}_m$ are tuples of variables and constants; $\{Z_1, \dots, Z_k\} = \{\bar{X}_1, \dots, \bar{X}_n\} \setminus \{\bar{Y}_1, \dots, \bar{Y}_m\}$. Note that if $m = n = 1$, the constraint will be reduced to a *referential constraint* as described in [25].

We show in [23] that instead of the standard referential expansion as described in [25], we need a *disjunctive referential expansion* to expand the query to a set of sub-queries which is semantically *equivalent* to the original query but includes the schema information. At the same time, each expanded sub-query is guaranteed to be *minimal* in the sense that it is not contained in any other sub-query in the set. The formal proof can be found in [23], we give here an example to illustrate how the integrity constraints affect the containment checking of queries in LDAP.

Example 7. Given the queries Q_1 and Q_2 where Q_1 asks the value of `uniqueId` of each entry that has no attribute `cn`, Q_2 asks also the value of `uniqueId` of all entries which are instances of the class `external`.

$Q_1 : q(X, \text{uniqueId}, Z) :- \text{sub}([o, \text{att.com}], X), \text{inst}(X, \text{uniqueId}, Z), \text{undef}(X, \text{cn}).$

$Q_2 : q(X, \text{uniqueId}, Z) :- \text{sub}([o, \text{att.com}], X), \text{inst}(X, \text{uniqueId}, Z), \text{inst}(X, oc, \text{external}).$

It is not hard to see that without the integrity constraints, Q_1 is not contained in Q_2 . However, if we first expand Q_1 using DRC1 above (in conjunction with the schema given in Table 1), the two sub-queries are obtained as follows:

$Q'_1 : q(X, \text{uniqueId}, Z) :- \text{sub}([o, \text{att.com}], X), \text{inst}(X, \text{uniqueId}, Z),$
 $\text{undef}(X, \text{cn}), \text{inst}(X, oc, \text{external}).$

$Q''_1 : q(X, \text{uniqueId}, Z) :- \text{sub}([o, \text{att.com}], X), \text{inst}(X, \text{uniqueId}, Z),$
 $\text{undef}(X, \text{cn}), \text{inst}(X, oc, \text{internal}).$

Note that semantically, we have $Q_1 =_{DRC} Q'_1 \cup Q''_1$, which means: any entry satisfies the query Q_1 must be the instance of either *external* or *internal*, or both. Consequently, we can test that $Q'_1 \subseteq Q_2$ and $Q''_1 \subseteq Q_2$ in the presence of the *implication constraint* given in Definition 7. The latter case should be explained: in Q''_1 the node should not have *cn* as attribute but it is an instance of *internal* which has *cn* as *required* attribute – which is a contradiction. Thus Q''_1 is unsatisfiable.

As a result, we get $Q_1 \subseteq Q_2$ in the presence of the *integrity constraints*, which could not hold otherwise. $\square$

Theorem 3. *The containment problem of LDAP queries with schema constraints is decidable.* $\square$

As shown in [25], the complexity of testing query containment of conjunctive queries in the presence of *referential constraints* and *implication constraints* is polynomially equivalent to that of conjunctive queries, which is Π_2^P -complete (with arithmetic comparison) [21].

4 Related Work

Semi-structured data models like XML have been intensively studied recently [2]. It is more difficult to give a relational semantics to the query language XPath, since its path expressions are not first-order expressible [1]. However, there have been several works on the relational semantics of the query languages (mostly a fragment of) XPath [1], XML-QL [17], and XQuery [20]. The early semi-structured data model SML can be transformed to *Datalog* with function symbols allowed [16]. Compared with the semi-structured data model, LDAP data model deploys the elegant *distinguished name* (DN) to express the path expression of entries of LDAP while at the same time keeping the first-order semantics.

The method of adding semantics of class schema constraints to the query processing first appears in [5] for the object-oriented data model. Disjunctions are applied to a single query to generate a set of *terminal queries* with class information for each object variable in the body of the query. However, the algorithm is more pragmatic and can be applied only for the object-oriented data model. Our method is the natural extension of the generalized integrity constraints and therefore can be applied more broadly.

In [1] disjunction (disjunctive embedded dependencies) is applied to the integrity constraints in the relational transformation of XPath. However, an extension of the traditional chase algorithm is applied which is different from our logical method.

[7] is the first paper concerning the query rewriting problem in LDAP. However, the data model and query model of LDAP are not built upon first-order

logic so that algorithms are pragmatic. We notice that query containment checking is the crucial point in dealing with query rewriting [9]. We believe that the results of this paper will be helpful in tackling the query rewriting problems in LDAP.

5 Conclusion

With the growing popularity of directories services in the World Wide Web, LDAP directory enabled networking is being promoted by companies including AT&T, Cisco, and IBM. Our focus on this paper, has been to present a first-order logic semantics of the LDAP query language. A complete transformation of LDAP queries to *Datalog* with generalized *integrity constraints* is given and the complexity of query containment of LDAP query model is analyzed with and without *integrity constraints* respectively.

Though independent of the implementation system, our results can be applied into any LDAP directory enabled systems. Actually, due to many benefits from using relational databases as storage systems for LDAP data, there have been implementations of LDAP that use the IBM DB2 database [18]. Our transformation of LDAP data model and query model provides the possibilities to cope with relational databases and LDAP enabled directories in a seamless platform in the data integration system.

The *integrity constraints* we analyzed in this paper can be used for query processing, semantic query optimization or semantic caching in LDAP systems. Furthermore, the problems of rewriting query using views which arise recently in the data integration system, can also substantially benefit from our results.

References

1. A. Deutsch and V. Tannen. Containment and Integrity Constraints for XPath Fragments. In *KRDB*, 2001.
2. S. Abiteboul, P. Buneman, and D. Suciu. Data on the web. Morgan Kaufmann, 2000.
3. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley Publishing Company, 1995.
4. S. Amer-Yahia, H. Jagadish, L. Lakshmanan, and D. Srivastava. On Bounding-schemas for LDAP Directories. Tech. Report, Concordia University, 1999.
5. E. P. F. Chan. Containment and Minimization of Positive Conjunctive Queries in OODB's. In *PODS'92*. ACM Press, 1992.
6. S. Chaudhuri and M. Y. Vardi. On the Equivalence of Recursive and Nonrecursive Datalog Programs. In *PODS'92*. ACM Press, 1992.
7. S. Cluet, O. Kapitskaia, and D. Srivastava. Using LDAP Directory Caches. In *PODS'99*. ACM Press, 1999.
8. P. Godfrey, J. Grant, J. Gryz, and J. Minker. Integrity constraints: Semantics and applications. In Jan Chomicki and Gunter Saake, editors, *Logics for Databases and Information Systems*. Kluwer, 1998.

9. A. Halevy. Answering Queries Using Views: A Survey. To appear in VLDB Journal, 2001.
10. T. Howes, M. Smith, and G. Good. Understanding and Deploying LDAP Directory Services. Macmillan Technical Publishing, 1999.
11. Innosoft. Innosoft's Resources for Directory Deployments. "http://www.innosoft.com/ldap_survey".
12. H. V. Jagadish, Laks V. S. Lakshmanan, Tova Milo, Divesh Srivastava, and Dimitra Vista. Querying Network Directories. In *SIGMOD'99*. ACM Press, 1999.
13. A. Klug. On conjunctive queries containing inequalities. In J. ACM 35:1, pp. 146–160, 1988.
14. A. Y. Levy and Y. Sagiv. Queries Independent of Updates. In *VLDB'93*, 1993.
15. P. J. Marron and G. Lausen. On Processing XML in LDAP. In *Proceedings of the VLDB'01*, 2001.
16. Y. Papakonstantinou. Query processing in heterogeneous information sources. PhD thesis, Dept. of Computer Science, Stanford University, 1996.
17. J. Shanmugasundaram, K. Tuftte, C. Zhang, G. He, D. J. DeWitt, and J. F. Naughton. Relational Databases for Querying XML Documents: Limitations and Opportunities. In *VLDB'99*, 1999.
18. S. S. B. Shi, E. Stokes, D. Byrne, C. F. Corn, D. Bachmann, and T. Jones. An enterprise directory solution with DB2. *IBM Systems Journal*, 39(2):360–383, 2000.
19. O. Shmueli. Decidability and Expressiveness of Logic Queries. In *PODS'87*. ACM, 1987.
20. I. Tatarinov, Zachary Ives, Alon Halevy, and Dan Weld. Updating XML. In *Proc. of ACM SIGMOD Conf. on Management of Data*, 2001.
21. R. van der Meyden. The Complexity of Querying Indefinite Information: Defined Relations. Ph.D thesis, Rutgers University, 1992.
22. M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3). Request for Comments 2251. <http://www3.innosoft.com/ldapworld/rfc2251.txt>, 1997.
23. F. Wei and G. Lausen. Conjunctive query containment in the presence of disjunctive integrity constraints. Submitted.
24. F. Wei and G. Lausen. Query Containment for Conjunctive Queries and Datalog with Negation. Technical Report. "<http://www.informatik.uni-freiburg.de/~fwei/paper/containment.ps>".
25. X. Zhang and Z. Meral Özsoyoglu. Implication and referential constraints: A new formal reasoning. *TKDE*, 9(6):894–910, 1997.

An XML Document Retrieval System Supporting Structure- and Content-Based Queries

Jae-Woo Chang

Dept. of Computer Engineering, Chonbuk National University
Chonju, Chonbuk 560-756, South Korea

jwchang@chonbuk.ac.kr

Abstract. In this paper, we design and implement an XML document retrieval system which can support both structure- and content-based queries. In order to support a structure-based query, we design four efficient index structures, i.e., keyword, structure, element, and attribute ones, and implement them by using the O2-Store storage system. In order to support a content-based query, we design and implement a high-dimensional index structure based on the CBF method so as to store and retrieve both color and shape feature vectors efficiently. Finally, we do the performance analysis of our XML document retrieval system in terms of system efficiency, such as retrieval time, insertion time, and storage overhead, as well as system effectiveness, such as recall and precision measures.

1 Introduction

In 1996, the XML (eXtensible Markup Language) was proposed as a standard markup language to make World Wide Web (in short Web) documents [1]. The XML has as good expressive power as SGML and is also easy to use like HTML. Recently, it has been common for users to acquire through Web a variety of XML documents. Meanwhile, because the number of XML documents is dramatically increasing, it is difficult to reach a specific XML document required by users. Moreover, an XML document not only has a logical and hierarchical structure commonly, but also contains its multimedia data, such as image and video. Thus, it is necessary to develop an XML document retrieval system that can support both the retrieval based on document structure and the retrieval based on image content.

In general, since the conventional XML document retrieval systems support only a structure-based query, it is impossible to efficiently deal with a user query which requires XML document retrieval based on both document structure and image content. In this paper, we design and implement an XML document retrieval system which can efficiently retrieve documents based on both document structure and image content. In order to support a structure-based query, we design four efficient index structures, i.e., keyword, structure, element, and attribute, by indexing XML documents based on the basic element unit and implement them by using the O2-Store storage system. For supporting a content-based query, we design and implement

a high-dimensional index structure based on the CBF method so as to store and retrieve both color and shape feature vectors efficiently.

This paper is organized as follows. In Section 2, we introduce related work in the area of structure-based and content-based document retrieval systems. In Section 3, we design an XML document retrieval system supporting structure- and content-based queries. In Section 4, we present the implementation and performance analysis of our XML document retrieval system. Finally, we draw conclusions and provide some issues for future research in Section 5.

2 Related Work

Because an element is a basic unit that constitutes a structured document (i.e., SGML or XML document), it is essential to support not only retrieval based on element units but also retrieval based on logical inclusion relationships among elements. Since there are a lot of studies on SGML documents, we, in this section, describe some related work on the representation of SGML document structures. First, RMIT in Australia proposed five query types for structure-based retrieval that should be supported in SGML information retrieval [2]. Most of the types consist of retrieval on upper-level elements (e.g., parent element), or on lower-level elements (e.g., child elements) from a given element. For supporting the five types of queries, RMIT proposed a *subtree model* which indexes all the elements in a SGML document and stores all the terms which are appeared in the elements [3]. Although the model supports efficient retrieval on a specific query, it has disadvantages of long indexing time and high storage overhead because index information should be repeatedly stored according to a tree depth. Secondly, RMIT proposed a *SCL structure* that extends the *GCL structure* [4]. After assigning numbers to both terms and markups in SGML documents, they use the *SCL structure* to store term interval, markups and inclusion relationships among elements. The *SCL structure* has an advantage that it can handle graph-structured documents, but it has two disadvantages that it requires a deletion operation and it cannot represent the depth of the elements effectively. Finally, SERI in South Korea proposed a *k-ary Complete Tree Structure* which represents a document as a k-ary complete tree [5]. In this method, each element corresponds to a node in a k-ary tree. Therefore, a relationship between two elements can be acquired by calculation. This method has an advantage that it is fast to find an element including a given logical relation by calculation. But, as the depth of a k-ary tree is deeper, the number of nodes is increasing exponentially with a large number of unused nodes. In the cases of partial insert and deletion, almost all of nodes should be changed in their assigned number.

There have been many researches on content-based retrieval systems for multimedia DBMS. The key issues of the studies include keyword extraction for text-based retrieval, image-processing techniques used for feature extraction of images, and multi-dimensional indexing techniques for fast retrieval, and content-based image retrieval based on color histogram, texture, and shape. First, *QBIC(Query By Image Content) project* [6] of IBM Almaden research center studied content-based image retrieval on a large on-line multimedia database. The study supports various query

types based on the visual image features such as color, texture, and shape. Secondly, the *Chabot project* [7] of UCB made use of a spatial indexing method on the relational database management system, POSTGRES, so as to perform simple attribute-based retrieval, text-based retrieval, and content-based image retrieval using color histogram. Thirdly, VisualSEEk of the Colombia University of USA is a developed database system as tool for content-based retrieval and browsing. It process user queries that combine a spatial location of image object and color [8]. Its purpose is an implementation of CBVQ(Content-Based Visual Query) which support efficient retrieval functionality for content-based database system. Finally, the *CORE (Content-base Retrieval Engine)*[9] of the National University of Singapore studied novel indexing techniques based on image features so that the engine provides content-based retrieval on multimedia objects. The engine provides the functionality of query feedback to support query refinement. The engine also uses the image features such as keyword, color, shape, texture so as to support visual browsing, similarity retrieval, fuzzy retrieval, and thesaurus-based text retrieval.

3 Structure-and Content-Based XML Document Retrieval System

A structure- and content-based XML document retrieval system mainly consists of five parts, such as a preprocessing part for parsing XML documents and doing image segmentation, an indexing part for generating index keys of the documents, a storage manager part for storing index information into a specific database, a retrieval part for integrating retrieved results into an unified one, and a user interface part for answering user queries by using Web browser. Figure 1 shows the whole system

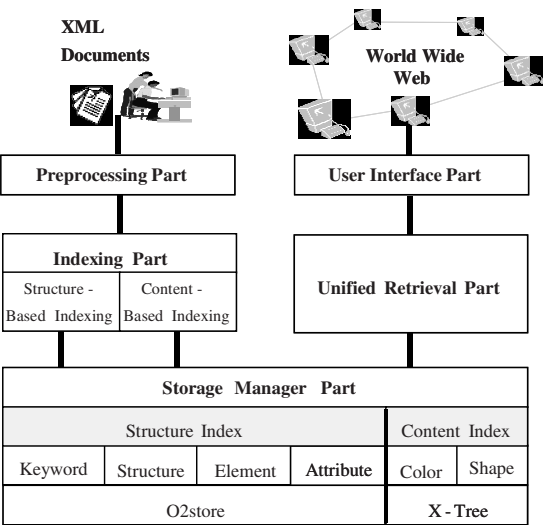


Fig. 1. System Architecture

architecture of our XML document retrieval system supporting structure- and content-based queries.

When an XML document is given, we first parse it and do image segmentation from it through the preprocessing part. The parsed document information is transported into the structure-based indexer in order to index its document structure consisting of element units. And the parsed image information is transported into the content-based indexer in order to get the index information of its color and its shape. The structure-based and content-based index information is separately stored into its index structures for structure-based retrieval and one for content-based retrieval, respectively. Using the index information extracted from a set of XML documents, some documents are retrieved by the retrieval part in order to obtain an unified result to answer user queries. Finally, the unified document result is given to users through a convenient user interface such as Web browser.

3.1 Indexing Based on Document Structure

Because an element is a basic unit for retrieving an XML document, it is required to support not only the retrieval based on document unit being used in the traditional document retrieval system, but also the retrieval based on element unit in a given depth. In addition, it is necessary to support a query on a logical inclusion between elements and on the characteristic value of elements. For this, we design an XML document retrieval system which efficiently supports element-based queries by constructing an index on a document structure after analyzing XML documents based on DTD. For instance, we suppose XML documents including Korean porcelain information with images. Figure 2 depicts a DTD grammar for representing its XML documents and one of its XML document instances. The XML document instance contains document structure between elements and attribute information. For example, the *porcelain* element has the *name* element as a child and the *decoration*

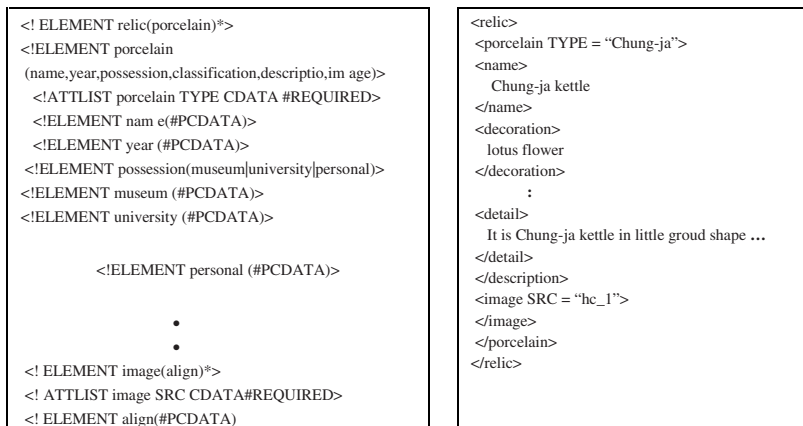


Fig. 2. A DTD grammar and an instance for XML documents

element has the *detail* element as a sibling. In addition, the attribute name of the *porcelain* element is ‘TYPE’ and its value is ‘CHUNG_JA’.

To make a document structure tree for XML documents, we first parse the XML documents by using sp-1.3 parser [10]. Next, we construct the document structure tree from the parsed result. Then the constructed tree is delivered into a low-level storage manager. Finally the storage manager extracts document structure information and image content information from the tree and stores them into a database. Figure 3 shows the constructed document structure tree.

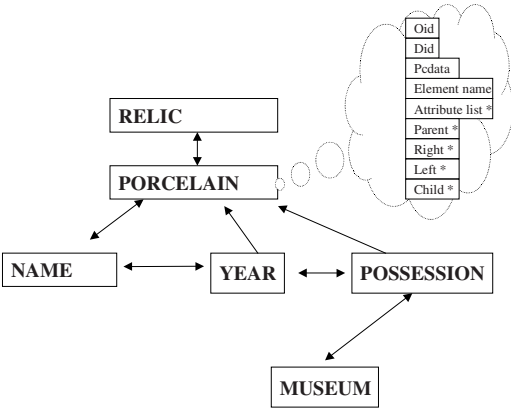


Fig. 3. Document structure tree and its element information

3.2 Indexing Based on Image Content

For image content-based retrieval, we analyze image objects of XML documents and extract image feature vectors by separating object regions from the background of images. To extract the object regions, we use the fuzzy c-mean (FCM) algorithm which is a generally famous clustering one to divide object regions from color images [11]. When we divide an image into two clusters, the FCM calculates the distances of a pixel from the center point of each cluster and assigns it to a cluster with shorter distance. It has an advantage that the separation of image objects from its background can be performed well when an image has little noise, as shown in our porcelain images.

In order to obtain an image feature vector for shape, we use the image object produced by the preprocessing and generate a 24-dimensional feature vector based on distances between the center point and a set of edge points. An algorithm for generating a shape feature is as follows. It can be directly applied to such applications as cyber museums, where an image has only a couple of salient objects

- i) After sorting each pixel of object in a column and a row respectively, calculate the central point of object using its maximum and minimum values.

- ii) By increasing 15 degrees at the central point, starting from the X-axis, select 24 pixel points met at the edge.
- iii) Compute the distance between the central point and the 24 pixel points on the edge.
- iv) Normalize the 24 distances by dividing them by the maximum distance.
- v) Generate a 24-dimensional feature vector.

Since the proximity among colors in the RGB color space doesn't mean their similarity among colors, we use HSV (Hue, Saturation, Value) color space model. This model provides a uniform distribution of colors and makes color transformation easy. In this model, H means an aggregate of color, ranging from 0 to 360 degree. S means the chroma of color, and V means the brightness of color. Here, we propose an algorithm to generate a 22-dimensional color feature vector as follows

- i) Transform all color pixels of an image object in the RGB color space into those in the HSV color space.
- ii) Generate a color histogram by using color histogram generation algorithm.
- iii) Normalize the color histogram by dividing it by the number of all the pixel.
- iv) Generate a 22-dimensional feature vector.

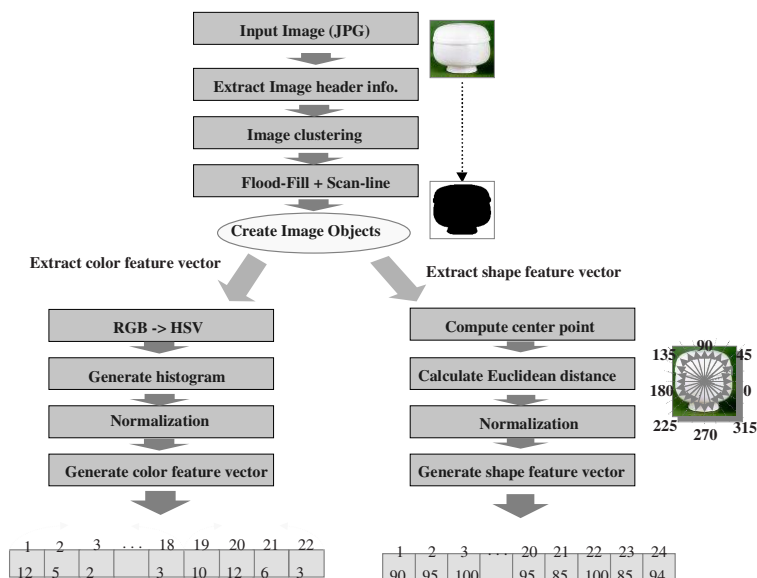


Fig. 4. Feature vector extraction of color and shape

Figure 4 shows an example in which a 24-dimensional feature vector and a 22-dimensional feature vector are extracted from the shape and the color of a real image, respectively.

3.3 Low-Level Storage Manager

For structured- and content-based queries, its low-level storage manager consists of index structures for structure-based retrieval as well as index structures for content-based retrieval. The index structures for structure-based retrieval are constructed by indexing XML documents based on element unit and consist of keyword, structure, element, and attribute index structures. Here, we will omit the element and the attribute index structure due to their simple structures.

3.3.1 Keyword Index Structure

The keyword index consists of three files, i.e., keyword index file being composed of keywords extracted from data token element (e.g., PCDATA, CDATA) of XML documents, posting file including the IDs of document and element where keywords appear, and location file containing the location of keyword appearance in elements. Figure 5 describes the keyword index structure and shows an extendible form of an index structure used for the conventional information retrieval system by adding an element information of a document.

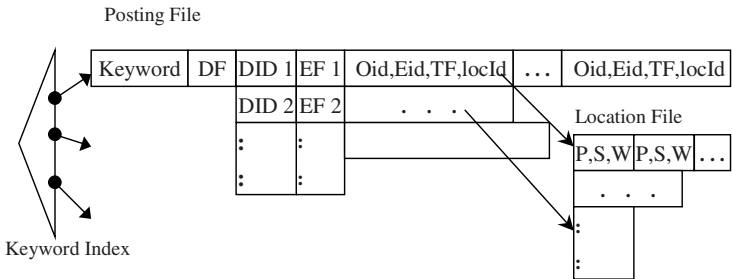


Fig. 5. Keyword index structure

DF (document frequency) represents the number of documents containing a given keyword. DID is the identifier of the document stored. Because a keyword can be appeared in a set of elements constituting a document, the index includes the different value of EF (element frequency) per each document where the EF means the number of elements having a given keyword in each document. Oid is the identifier of the element stored. Eid is an identifier for an element name. Here we use Eid (element name ID) rather than the actual element name because the element name being variable in size is appeared repeatedly. TF (term frequency) represents the number of keyword occurrences in an element. locId is the identifier of location information, i.e. paragraph (P), sentence(S), and word(W), in the location file. Since the location information is only used to answer a keyword adjacency query, we can access only the posting file except the location file to answer a general query, leading to fast response time.

3.3.2 Structure Index Structure

The structure index is used for searching an inclusion relationship among elements. So, it should represent the logical structure of a document and should guarantee good performance on both retrieval time and storage overhead. In this paper, we propose an element unit parse tree structure to represent the hierarchical structure of a document. In the structure, we easily find an inclusion relationship among elements because an element contains the location of its parent, its left sibling, its right sibling, and its first left child. Figure 6 shows the element unit parse tree structure of a given parse tree.

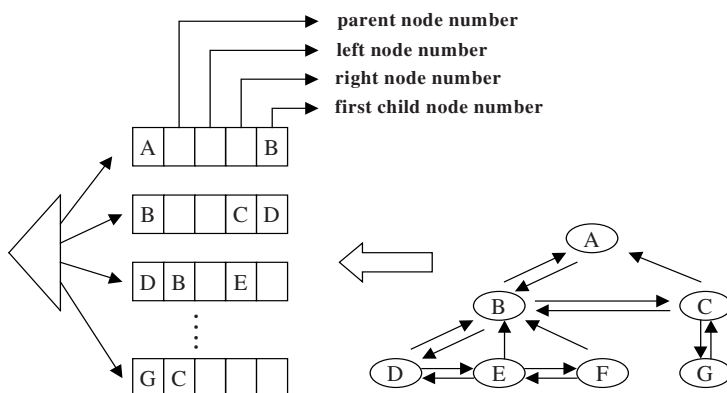


Fig. 6. Element unit parse tree structure

Figure 7 shows the structure index structure based on the element unit parse tree where an element is identified by Oid and Eid. For fast searching of inclusion relationship among elements, we make use of the identifier of parent element (ParentOid), the identifier of left sibling element (LeftOid), the identifier of right sibling element (RightOid), and the identifier of the first left child element (FchildOid). In addition, NW(node weight) represents relevance weighting degrees between a parent node and one of its child nodes. It is computed as the similarity between a parent term vector and its child term vector.

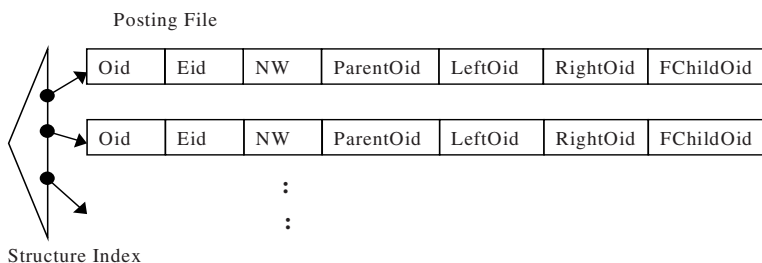


Fig. 7. Structure index structure

3.3.3 Index Structure for Content-Based Retrieval

The index structure for content-based retrieval is a high-dimensional index structure based on the CBF method so as to store and retrieve both color and shape feature vectors efficiently. A main focus on managing a large number of XML documents is retrieval performance. As the number of dimensions of feature vectors is increasing, the retrieval performance of the traditional index structures is exponentially increasing. To cope with this problem, the CBF method [12] was proposed as an efficient high-dimensional index structure to achieve good retrieval performance even though the dimension of feature vectors is high. The CBF divides a high-dimensional feature vector space into cells. It performs filtering based on a distance between an object feature vector and the center of a cell including it, in addition to filtering based on cell signatures before accessing a data file. Since our index structure is made based on the CBF, it can support various types of queries, like point, range, and k-nearest neighbor search. To achieve better retrieval performance on the high-dimensional feature vectors of both color and shape, we are doing research on a parallel CBF method.

3.4 Unified Retrieval

Even though there have been various retrieval models in the traditional text-based information retrieval, there is little research on retrieval models for structured- and content-based XML information retrieval. To answer a document structure query, a similarity (S_w) between an element q and an element t is computed as the similarity between the term vector of node q and that of node t [13], as shown in the following equation.

$$S_w = \text{COSINE} (NODE_q, NODE_t) = \frac{\sum_{k=1}^m (TERM_{qk} \cdot TERM_{tk})}{\sqrt{\sum_{k=1}^m (TERM_{qk})^2 \cdot \sum_{k=1}^m (TERM_{tk})^2}}$$

When a result to answer user query is a set of documents, the documents can be represented as $D = \{ E_0, E_1, \dots, E_{n-1} \}$ where E_i means an element i in a document D . Also, a similarity (D_w) between an element q and a document D is computed as follows.

$$D_w = \text{MAX} \{ \text{COSINE} (NODE_q, NODE_{E_i}), 0 \leq i \leq n-1 \}$$

To answer an image content query, we first extract color or shape feature vectors from a given query image. Then we compute Euclidean distances between a query color (or shape) feature vector and the stored image color (or shape) feature vectors by searching the color (or shape) index. Finally we compute a similarity between the query feature vector and the image feature vector as $1 - (\text{Euclidean distance} / \text{maximum distance})$, and retrieve relevant documents with high similarity in the decreasing order of the similarity. In case we require content-based retrieval using both color and shape feature vectors, we compute a unified similarity by the multiplication of both the color and the shape similarities. A similarity, $C_w(q, t)$,

between a query image q and a target image t in the database is calculated as the following equation. Here $Distc(q, t)$ and $Dists(q, t)$ mean a color vectors distance and a shape vector distance between a query image q and a target image t , respectively. Nc and Ns mean the maximum color and the maximum shape distances for normalization, respectively.

$$C_w = \begin{cases} 1 - \frac{Distc(q, t)}{Nc}, & \text{if a query contains only a color feature.} \\ 1 - \frac{Dists(q, t)}{Ns}, & \text{if a query contains only a shape feature.} \\ (1 - \frac{Distc(q, t)}{Nc}) \times (1 - \frac{Dists(q, t)}{Ns}), & \text{if a query contains both color and shape feature.} \end{cases}$$

Finally, when α is the relative weight of the structured-based retrieval compared with the content-based retrieval, a similarity (T_w) for both structured- and content-based composite query is calculated as the following equation. If the weight of the structured-based retrieval is equal to that of the content-based retrieval, α is 0.5.

$$T_w = \begin{cases} C_w \times \alpha + D_w \times (1 - \alpha), & \text{if results are document for user query} \\ C_w \times \alpha + S_w \times (1 - \alpha), & \text{if results are element for user query} \end{cases}$$

3.5 User Interface

We construct 630 XML documents with a proper XML DTD format from a porcelain book published in Korean National Central Museum. In order to design an efficient structure- and content-based document retrieval system, we classify queries for retrieving XML documents into two types, i.e., simple query and composite query. The simple query can be divided into keyword-based, structure-based, attribute-based, and content-based queries. The examples of the simple query are as follows

- ☐ Keyword-based : Find documents which contain 'Buddhist image' term.
- ☐ Structure-based : Find all children elements of [Porcelain] element.
- ☐ Attribute-based : Find documents whose attribute type is 'chong-ja'.
- ☐ Content-based: Find documents or images having a specific color and shape.

The composite query is the composition of simple queries. The figure 8 shows a user query interface. It consist of five parts for making a user query, that is, structure-based, keyword-based, attribute-based, color-based and shape-based queries. For example, a user wants to retrieve documents that have 'Buddhist image' keyword in all children element of [porcelain] element whose TYPE attribute value is 'chong-ja' and which include an image whose color is 'blue' and whose shape is like 'kettle'. The figure shows result for the query and its similarity with images in a database for the combination of structure- and content-based queries. When a user wants more detailed information, another window can browse the detail of the result.

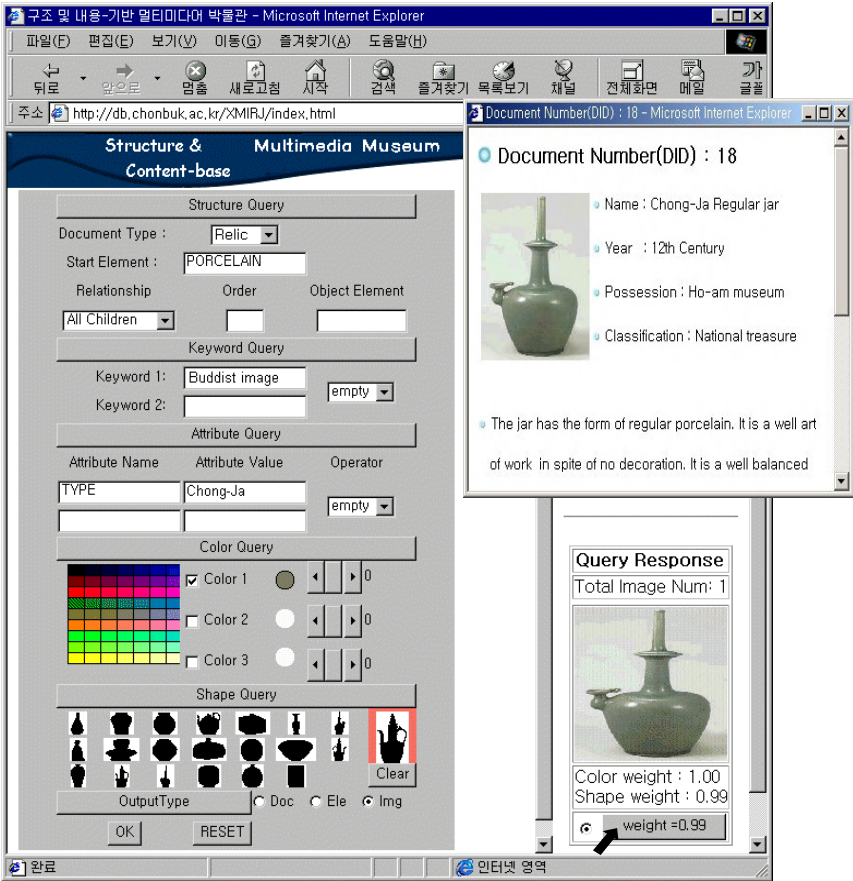


Fig. 8. User query interface

4 Implementation and Performance Analysis

We implement our structure- and content-based XML document retrieval system under SUN SPARCstation 20 by using GNU CCv2.7 compiler. For this, we make use of O2-Store v4.6 [14] as a storage system and Sp-1.3 as an XML parser. To evaluate our system efficiency, we measure retrieval time, insertion time and storage overhead. Table 1 shows a test data set used for the performance analysis.

Table 1. A test data set

The number of Documents	630 XML documents
Document average size	1.20K+42K(image)
The number of elements	10754
The number of keyword	37807

For insertion time, it takes less than 1 second to insert an image content information into color and shape index. Meanwhile, it takes about 6 seconds to insert the structured information of XML documents into keyword, attribute, structure, and element index structures. Figure 9 shows retrieval times for queries base on keyword, attribute, structure, image color, and image shape. The retrieval times for the color- and the shape-based queries are less than 1 second, respectively. The retrieval times for keyword- and attribute-based queries are about 2 seconds, respectively, and that for the structure-based query is 6.5 seconds. Consequently, it is shown that the time for answering the structure-based query is the longest.

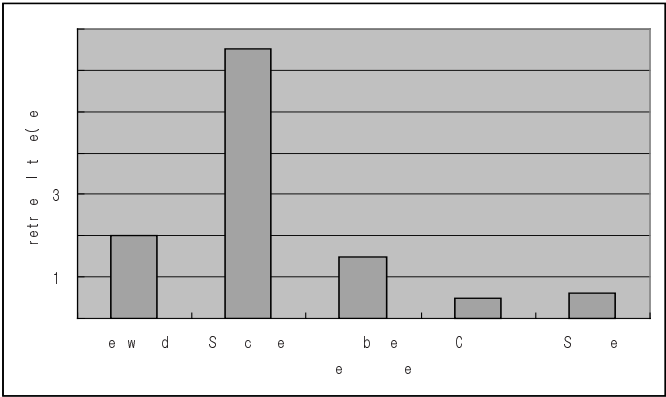


Fig. 9. Retrieval time for simple queries

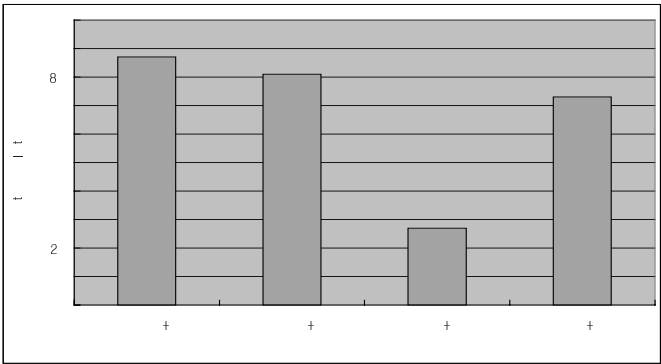


Fig. 10. Retrieval time for complex queries

Figure 10 shows retrieval times for complex queries, such as structure + keyword, structure + attribute, keyword + color and structure + shape queries. The retrieval time for the combination of structure- and keyword-based queries is less than 9

seconds and the combination of structure- and attribute-based queries is about 8 second. The retrieval time for the combination of structure- and shape-based queries is about 7 second and that for the combination of keyword- and color-based queries is less than 3 second. Consequently, it is shown that a complex query excluding the structure type requires the shortest time to answer it. We measure storage overhead as a ratio of the total size of our index files to that of the original XML documents. It is shown that our XML document retrieval system requires about 1200% storage overhead.

To evaluate retrieval effectiveness, we measure recall and precision [13] with a test group consisting of 10 graduate students from the Computer Engineering department of Chonbuk National University, South Korea. Table 2 shows the recall and precision measures of our XML document retrieval system when we find the most similar K porcelains as the most similar ones. As K is increasing, the precision is decreasing while the recall is increasing. In case of K=10, the recall is 0.6 and the precision is about 0.4 for the color-based query, while the recall is about 0.6 and the precision is about 0.3 for the shape-based query.

Table 2. Precision and recall measures

	Color		Shape	
	Precision	Recall	Precision	Recall
K = 7	0.64	0.33	0.57	0.23
K = 10	0.60	0.37	0.55	0.32

Our system can support both structure- and content-based queries, whereas other systems in the related work do not support both of them. Thus, it is not reasonable to make an overall comparison of our system with other systems. Due to the significance of structure-based queries, we, in this paper, compare our system with the *k*-ary *Complete Tree Structure* [5] witch is known as a promising structure for the structure-based queries. The retrieval time for the structure-based query is 6.5 seconds, while that for the *k*-ary tree is 6.3 seconds. This shows nearly the same performance for structure-based queries.

5 Conclusions

In this paper, we designed and implemented an XML document retrieval system that can efficiently retrieve documents based on both document structure and image content. In order to support efficient structure-based queries, we designed our keyword, structure, element, and attribute index structures by indexing XML documents based on the basic element unit and implemented them by using the O2-Store storage system. For image content-based queries, we designed and implemented a high-dimensional index structure based on the CBF method for both color and shape feature vectors. In our XML document retrieval system, the retrieval time for the structure-based query is about 3 seconds while that for the content-based query is below 1 second. Also, the retrieval time for the combination of structure- and keyword-based queries is shown to be the longest, i.e. over 8 second. Our XML

document retrieval system spends about 6 seconds for inserting a document and requires about 50 % storage overhead. As a further research, in order to make our XML document retrieval system popular, it is necessary to implement it by using a public storage system.

Acknowledgement. This work was supported in part by grant No. 2001-1-51200-006-2 from the Basic Research Program of the Korea Science & Engineering Foundation.

References

- [1] eXtensible Markup Language(XML), <http://www.w3.org/TR/PR-xml-971208>.
- [2] R. Sack-Davis, T. Arnold-Moore and J. Zobel, "Database Systems for Structured Documents," In Proc. Informational Symposium on Advanced Database Technologies and Their Integration, pp 272-283, 1994.
- [3] B. Lowe, J. Zobel and R. Sacks-Davis, "A Formal Model for Databases of Structured Text," In Proc. the 4th International Conference on Database Systems for Advanced Applications, pp 449-456, 1995.
- [4] T. Dao and R. Sacks-Davis, and J.A. Thom, "An Indexing Scheme for Structured Document and its Implementation," In Proc. the 5th International Conference on Database Systems for Advanced Applications, pp 125-134, 1997.
- [5] S.G. Han, et. al., "Design and Implementation of a Structured Information Retrieval System for SGML Documents," In Proc. the 6th International Conference on Database Systems for Advanced Applications, pp 81-88, 1999.
- [6] M. Flickner, et. al., "Query by image and video content: The qbic system," IEEE Computer, Vol. 28, No.9, pp. 23-32, 1995.
- [7] V.E. Ogle and M. Stonebraker, "Chatbot: Retrieval from a relational database of images," IEEE Computer, Vol. 28, No.9, pp 40-48, Sep. 1995.
- [8] J. R. Smith and S. F. Chang, "VisualSEEk: a Fully Automated Content-Based Image Query System," ACM Multimedia Systems, Vol. 4, Nov 1996.
- [9] J.K. Wu, et. al., "Core: a content-based retrieval engine for multimedia information systems," ACM Multimedia Systems, Vol. 3, No. 1, pp 25-41, 1995.
- [10] <http://www.jclark.com/sp>.
- [11] J.C. Bezdek and M.M. Trierdi, "Low Level Segmentation of Aerial Image with Fuzzy Clustering," IEEE Trans. on SMC, Vol.16, pp. 589-598, 1986.
- [12] S.G. Han and J.W. Chang, "A New High-Dimensional Index Structure using a Cell-based Filtering Technique," In LNCS 1884 (ADBIS-DASFAA 2000), pp 79-92, 2000.
- [13] G. Salton and M. McGill, "An introduction to Modern Information Retrieval," McGraw-Hill, 1983.
- [14] Deux et al. "The O₂ System," IEEE Communication of the ACM, Vol. 34, No. 10, 1991.

Extraction of Partial XML Documents Using IR-Based Structure and Contents Analysis

Kenji Hatano¹, Hiroko Kinutani¹,
Masatoshi Yoshikawa^{1,2}, and Shunsuke Uemura¹

¹ Nara Institute of Science and Technology
8916-5 Takayama, Ikoma 630-0101, Japan
{hatano,hiroko-k,yosikawa,uemura}@is.aist-nara.ac.jp

² National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, Japan

Abstract. As Internet technologies develop, XML is becoming widely used as a standard data/document format. Although the use of XML documents has attracted public attention, the application of IR technologies in XML document retrieval is still in its premature stage. We foresee that typical XML queries for end-users will be very terse, like those used with current Web search engines. Therefore, an XML search engine should be able to search appropriate retrieval results using only a few keywords. In this paper, we introduce a notion of context nodes. Context nodes are used to automatically extract coherent partial documents without the knowledge of XML document structures. This method is useful because it does not require domain analysts to analyze DTDs and specify candidate partial documents beforehand. We use the term “context search” to represent search methods which employ the notion of context node. As an instantiation of context search methods, we have developed algorithms to identify result partial documents in the vector space model. We made a performance evaluation to verify the effectiveness of our method.

1 Introduction

Recently, XML (Extensible Markup Language) [2] has attracted much attention as a universal data format on the Web. XML are now being used in many application domains. In near future, various kinds of data and documents will be expressed in XML and will be made available on the Web. This situation is very similar to the current HTML documents. Accordingly, XML search engines will become very important as an end-user’s tool for exploration of the Web.

Until now, many XML query languages have been proposed [1]. XQuery [5], a new query language proposed by W3C, adopts other languages’ advantages. Query languages for XML have an essential importance in specifying complex queries declaratively. However, to use XML query languages, users must have precise knowledge about the structure of target XML documents beforehand, and specify the selection condition and the structure of the retrieval results. Such XML query languages are clearly not suitable for the user interface in

XML search engines. Given that a large number of XML documents which have various structures are collected in search engines, understanding the full logical structures of such a heterogeneous set of documents is not possible. This implies XML search engines cannot expect users to utilize logical structures of XML documents in specifying search conditions or target document structures. Therefore, we foresee a much simpler form of queries for end-users, resembling those in use with current Web search engines, in which typical queries consist of a few keywords [12]. Queries consisting of a few keywords against heterogeneous XML documents are inherently ambiguous. Hence the search engine should rank the relevance of retrieved documents.

Other problems with XML query languages is that they do not have a function to express ambiguity, or to rank retrieval results as in conventional IR techniques. Moreover, because users cannot specify the structure of result documents, system should retrieve partial documents of appropriate granularity as the result of retrieval, in light of the fact that some XML documents might be very lengthy, and only a sub-part of documents might be relevant to the user's query.

In this paper, we discuss document retrieval in XML search engines. We propose methods to retrieve relevant XML partial documents utilizing both the document structure and the occurrence or frequency of keywords.

2 Related Work

The application of IR techniques in searching XML documents has become a challenging area in recent years, because there are wide range of assumptions on queries and retrieval results.

Some researchers have proposed an extension of XML query languages by adding the functionality of full text string matching. Florescu et al. proposed a method which integrates a "keyword search" into "XML query processing" [9]. This research is based on the premise that a user does not have knowledge of document structure beforehand, though the user does specify the elements to be extracted. XML Query Engine [13] also extended XQL [17] by adding text retrieval operators. These researches do not consider the ranking of retrieved documents. Moreover, they cannot search retrieval results if users do not specify the document structure.

Some researchers have considered not only the traditional IR-style similarity conditions, but also the structural (or contextual) similarity conditions. XXL [21], proposed by Theobald et al., is an SQL style query language in which queries can be specified without distinguishing element type names, attribute names and document contents. They have also studied how to combine an XML graph pattern matching and relevance estimations. XXL assumes that users have at least partial knowledge on document structures. XIRQL [7,8] is an extension of XQL, and was designed to use IR techniques. Using this query language, a user can specify weights to nodes so that retrieval results are influenced by the weights. As for the retrieved partial documents, domain analysts specify root

nodes for partial documents which are adequate as query results. Their goal is that the system should be able to automatically determine a suitable context for the user's query.

Some venture capital organizations have developed XML search engines. XYZFind [6] is a native XML database with radical "universal schema" architecture. It does not require users to have prior knowledge on schemas. Initially, users issue keywords as a query as well as current Web search engines, and then the system retrieves schemas of documents which contain all of the keywords. In this manner, the system supports users to find proper schemas interactively. Once users found proper schemas, the system shows a search form specific to the schema. The interactive approach would not be feasible when a large number of schema is stored in the search engine. Another problem is that filling the search form is not easy when the schema is huge.

The purposes of these XML search engines are aligned with that of our research from the viewpoint of obtaining suitable retrieval results using a few query keywords. However, there are significant differences between these search engines and our research. Firstly, both XIRQL and XYZFind are assuming the existence of schemas, and based on the schema. Our approach does not require schemas. Secondly, these search engines cannot extract result partial documents automatically. In XIRQL, the domain analyst has to define the root of result partial documents beforehand. Though the authors have mentioned the necessity of retrieval of partial documents in [6], no concrete algorithm has shown. To our knowledge, excepting our approach, there is no current research on extracting partial documents automatically without the knowledge of schemas.

3 XPath Data Model

In this paper, we follow the notations and the data model defined in XPath [4]. In the XPath data model, an XML document instance is modeled as a directed tree. Among the seven types of nodes in the data model, we limit our attention to a root node, element nodes, attribute nodes and text nodes for the sake of simplicity¹. In an XML tree, leaf nodes are text nodes or attribute nodes, and intermediate nodes are element nodes. The child element node of the root node is called *document node*. In the XPath data model, somewhat strange parent/child relationship between element nodes and attribute nodes is used. Even an element node is a parent of an attribute node, the attribute node is not a child of the element node. In this case, we regard the attribute node is a child of the element node. This is the only difference between the XPath data model and our data model. The *expanded-name* of an element node (an attribute node, respectively,) is the element type name (the attribute name, respectively,) of the node. The *string-value* of a text node is text itself². The string-value of

¹ The remaining three types of nodes are namespace nodes, processing instruction nodes and comment nodes.

² In this paper, we do not consider hyperlinks in one document or among two or more documents.

an attribute node is the value of the attribute. The string-value of an element node is the concatenation of the string-value of all text nodes descendants of the element node. Fig. 1 shows a logical structure of a sample XML document instance. There are two kinds of structured document retrieval models: one is

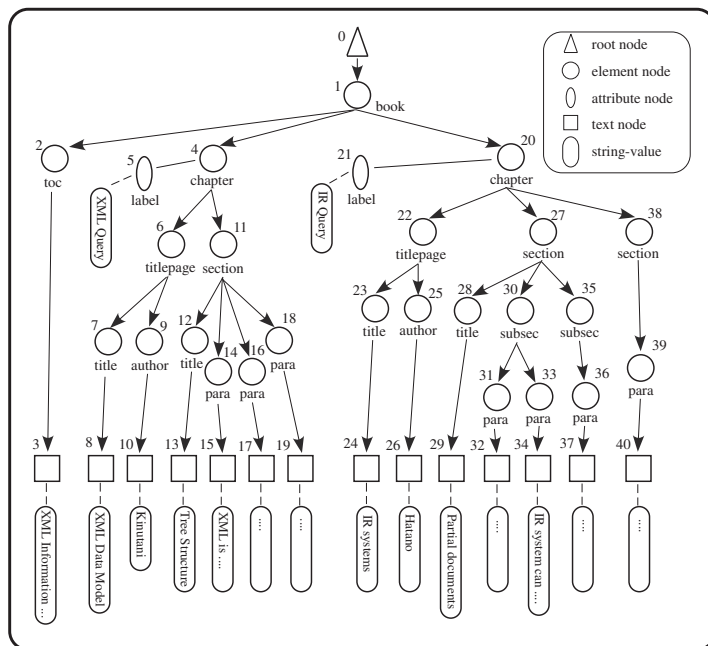


Fig. 1. A Tree Representation of an XML Document Instance.

based on non-overlapping lists and the other is based on proximal nodes [3]. Our model on partial documents is similar to the model based on proximal nodes [16]. That is, we adopt a logical model of XML partial documents as a subtree whose root node is an element node. Therefore, we can identify a partial document by the reference number, say n , of the root node of the partial document tree, and we call such partial document as “partial document $\#n$.”

4 Identification of Candidate XML Contextual Partial Documents

The process of our retrieval approach, the context search, consists of the following two steps:

1. identification of XML partial documents which are coherent and meaningful unit; and

2. evaluation of the relevance of the identified partial documents against queries.

In this section, we focus our attention to the first step. The second step is presented in section 5.

Let us consider the case when an end-user issued a single keyword query “Hatano.” Which partial documents are relevant as an answer of this query? The minimum partial document containing a character string “Hatano” is the partial document #25. A text representation of this partial document is shown below:

`<author>Hatano</author>`

We consider, however, this partial document is not informative enough for end-users, because he/she could not know what ‘Hatano’ has authored. Obviously, the other end of the line, i.e. returning the whole document, is not adequate either. Because the document in Fig. 1 has two chapters, and ‘Hatano’ is the author of the second chapter, the partial document #20 will be most relevant as an answer. That is, we regard the partial document #20 as a semantically consolidated granule of documents.

We call such semantically consolidated granule of documents as *CS (Coherent Subdocument)*. We regard CSs are adequate partial documents as retrieval results. As the above example suggests, not every partial document could be regarded as a CS. Some partial documents might be too small or too large as a granule of CS. We use the term “*context search*” to represent search methods that return CSs as an answer of a user’s query. *Context search* is an approach that automatically identifies CSs without the knowledge of DTDs (Document Type Definitions).

The problem we address in this section is the identification of CSs in a given XML document.

4.1 Manual Analysis of DTDs

The first possible approach to the identification problem is the use of DTDs if they are available. This approach was adopted in XIRQL. We call this approach *selected-node*.

selected-node approach: This approach requires domain analysts’ manual analysis of DTDs. If XML documents are complied with DTDs which are available to a domain analyst of the search engine, he/she can analyze the DTDs and select out appropriate elements as a root of CSs.

Example 1 Fig. 2 shows a sample DTD. The XML document in Fig. 1 is complied with this DTD. From the analysis of the DTD, a domain analyst might selects out `<chapter>`, `<section>`, and `<subsec>` as elements type names of root nodes of CSs. This means CSs are #4, #11, #20, #27, #30, #35 and #38 in Fig. 1.

One of the advantages of the *selected-node* approach is that it does not require the scanning of document instances.

```

<!ELEMENT book      (toc, chapter*)>
<!ELEMENT toc       (#PCDATA)>
<!ELEMENT chapter   (titlepage, section*)>
<!ELEMENT titlepage (title, author)>
<!ELEMENT section   (title?, (subsec* | para*))>
<!ELEMENT title     (#PCDATA)>
<!ELEMENT author    (#PCDATA)>
<!ELEMENT subsec    (para*)>
<!ELEMENT para      (#PCDATA)>
<!--ATTLIST label   (#IMPLIED)-->

```

Fig. 2. A Sample Document Type Definition (DTD).

4.2 Automatic Identification of Context Nodes

If DTDs are available, the *selected-node* approach will give semantically adequate CSs. However, DTDs are not always available for domain analysts. For example, DTDs are not available for XML documents which are dynamically created from databases at remote sites. Furthermore, even DTDs are available, manually analyzing large number of DTDs is not feasible.

When we use well-formed XML documents as the target of information retrieval, we can consider two methods to manage XML document instances. One is schema extraction from XML document instances, the other is making indices from XML document instances. Goldman et al. proposed a method to extract schema from XML document instance without DTD [10,11]. However, there is a problem that it is difficult to extract common structure from the XML document instances if there are some kinds of schema in each XML document instance. That is, the use of a method to extract common schema from well-formed XML documents and to specify retrieval results is limited. Considering these points, we have decided to use the latter approach, a method to extract all candidates of partial documents from XML document instances and make their indices for information retrieval.

In this section, we present an approach to automatically identifying CSs without the knowledge of DTDs. This approach is based on a heuristic algorithm obtaining CSs of a well-formed XML document from the document structure. We have proposed a method identifying result partial documents for search conditions based on document structures [14]. The method is applicable to identification of CSs in XML search engines. Of course, in general, the document structure does not fully convey the semantics of the document. However, as demonstrated in Section 5, our heuristic algorithm gives good performance.

The key idea of our approach is to define the *context node* for text nodes and attribute nodes. The reason we restrict our attention to text nodes and attribute nodes is that only nodes of these two types directly have string-value. The context node of a node n is an element node which is an ancestor of n . Intuitively, the context node gives the boundary of the context of a text node or an attribute node. The context node is defined to be in elder generation as possible so that the node n plays a unique role in the partial document defined by the context node. For example, in Fig. 1, the context node of the text node

#26 is the element node #20. This is because every node between the path (#26, #22) does not have a sibling node with the same expanded-name. However, the node #20 has a sibling node with the same expanded-name (i.e. the node #4). This implies the role of the text node #26 is unique within the partial document #20, but not unique within the partial document #1. In fact, we can observe the node #26 represents the author of the second chapter (#20), while the node #10 represents the author of the first chapter (#4). As another example, let us consider the text node #32. The parent node #31 has the sibling node #33 with the same expanded-name. If we apply the rule explained in the above example, the context node of #32 would become #31. However, we consider the node #31 does not give a proper boundary of the context of #32. To avoid such cases, we ignore the sibling nodes of the parent node. To find the context node of the node n , we start from the grandparent of n , and go up until we find a node having a sibling with the same expanded-name. Hence, the context node of #32 is defined to be #30. The formal definition of context nodes follows:

Definition 1 (Context node) *For a text node or an attribute node n in an XML document D , the context node of n in D is denoted by $\mathbf{context}(n)$, and defined as follows:*

1. *For an attribute node n , $\mathbf{context}(n)$ is the parent element node of n .*
 2. *For a text node n , $\mathbf{context}(n)$ is defined as follows: Let $g(n)$ be the grandparent node of n . Then, $\mathbf{context}(n)$ is the lowest node m on the path between $g(n)$ and the document node n_d such that m has a sibling node having the same expanded-name with m .*
- If such a node m does not exist, $\mathbf{context}(n)$ is defined to be n_d .*

The definition of context node is based on the topology of document trees, especially on the number of siblings with same expanded-name. This number is restricted by the occurrence indicators (*, +, or ?) in the element type declaration of DTDs if they are available. Since we do not use DTDs, the number of siblings we use in the definition of context nodes is the one actually found in document instances. Identification of context nodes is easily done by scanning XML document instances.

Next we present the approach to identifying CSs based on context nodes.

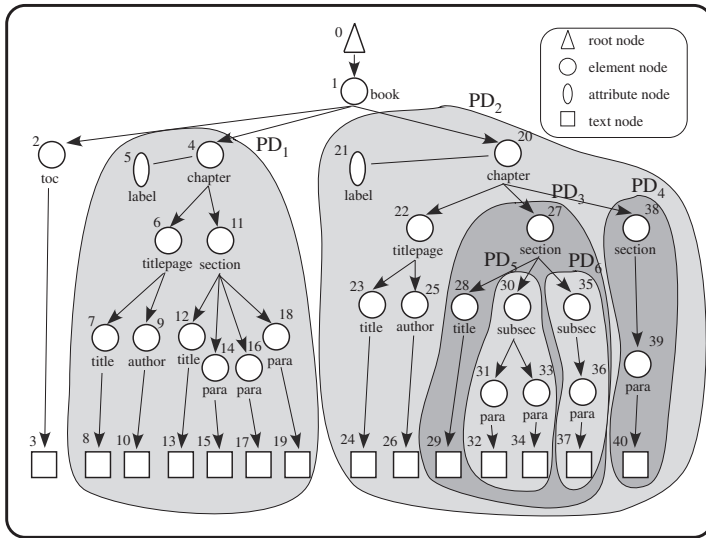
context-node approach: The CSs of a (well-formed) XML document D is defined as partial documents rooted by $\mathbf{context}(n)$, where n is a text node or an attribute node of D .

Example 2 *Table 1 summarizes the context nodes of text nodes and attribute nodes in the document tree in Fig. 1. Following the context-node approach described above, Fig. 3 shows the identified CSs. We have six context nodes (#4, #20, #27, #30, #35 and #38) excepting the document node. PD_i ($i = 1, 2, \dots, 6$) as shadow areas in Fig. 3 are those identified CSs. Note that, in this example, identified context nodes' element names are "chapter", "section" and "subsec."*

Table 1. Context nodes of text nodes and attribute nodes.

context node	#1	#4	#4	#4	#4	#4	#4	#20	#20	#27	#30	#30	#35	#38
text node	#3	#8	#10	#13	#15	#17	#19	#24	#26	#29	#32	#34	#37	#40

context node	#4	#20
attribute node	#5	#21

**Fig. 3.** An Example of Context Node.

5 Experimental Evaluation of Partial Document Extraction in the Vector Space Model

In this section, we show how result partial documents could be identified in the vector space model [18].

In [14], we adopted the boolean model to search retrieval results, so that we cannot get them with rank. However, by adopting the vector space model, we can compute the retrieval results' similarities between partial documents and a user's query. We regard each CS as a candidate result partial document. In our system, we adopt the *context-node* approach described in section 4 to extract CSs. In order to verify the effectiveness of the *context search* approach, we have performed experiments comparing *context search* and two other methods: the *selected-node* approach presented in section 4, and the *all-node* approach explained below:

all-node approach: This is a naive decomposition method. In this method, a document is decomposed into partial documents the root node of which is an element node except the document node. In this method, the number of CSs becomes the number of element type nodes minus one.

Note that the *context-node* approach and the *all-node* approach do not require the domain analyst’s intervention to extract CSs, while the *selected-node* approach requires domain analyst’s manual analysis.

5.1 Our Retrieval System Setup

In this subsection, we explain the processes of our retrieval system in detail. Fig. 4 shows a sample flow of the processing in our prototype system. As shown in Fig. 4, our prototype system is separated into three modules; extracting partial documents, making an index of the partial documents, and ranking the partial documents.

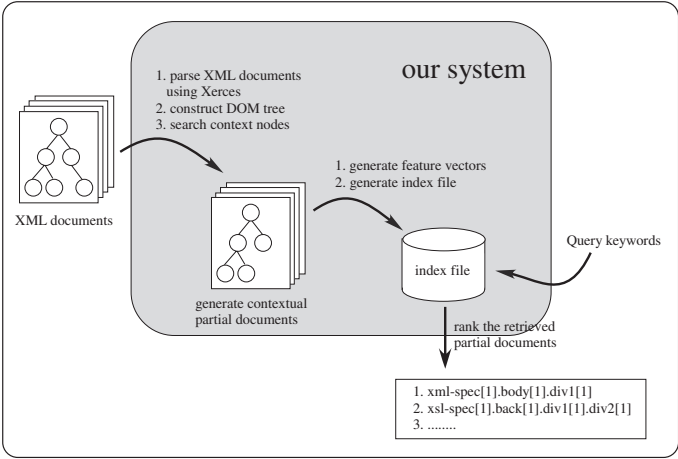


Fig. 4. The Flow of Processing in Our System.

1. **Extracting Partial Documents from Original XML Documents**
First, our system parses XML documents using an XML parser called Xerces³, and constructs DOM trees [15] from these documents. Then, it extracts CSs by analyzing the DOM trees.
2. **Generating an Index of the Partial Documents**
In our system, the similarity between a query and a CS is calculated from the similarity of the query and the CS’s self-or-descendant element nodes which has a text node or an attribute node as a direct child. Let e_j ($j = 1, 2, \dots, N$) be an element node which has a text node or an attribute node as a child. Our

³ <http://xml.apache.org/xerces-j/index.html>

system calculates the frequency of the terms t_k ($k = 1, 2, \dots, n$) appearing in all XML documents, and generate feature vector $\mathbf{F}(e_j)$ of e_j . The feature vector $\mathbf{F}(e_j)$ is calculated using a keyword-weighting strategy called tf-idf schemes [19] as follows:

$$\mathbf{F}(e_j) = (w_{t_1}^{e_j}, w_{t_2}^{e_j}, \dots, w_{t_n}^{e_j}) \quad (1)$$

where

$$w_{t_i}^{e_j} = \frac{tf(t_k, e_j)}{\sum_{j=1}^N \sum_{k=1}^n tf(t_k, e_j)} \cdot \log \frac{N}{ef(t_i)}$$

$tf(t_k, e_j)$ is term frequency appearing the term t_k in element e_j , and $ef(t_k)$ is element frequency appearing the term t_k in all elements. Then our system generates an index of all feature vectors of the element nodes. To realize high-speed vector search, we adopt the inverted file technique. In this technique, our system just calculates similarities of the documents which include query keywords, so that we can reduce the calculation cost of our system.

3. Ranking the CSs in Order of Their Scores

When a user inputs a query consisting of keywords into our system, our system generates a query vector $\mathbf{q}$. The base of the query vector is the same as that of the element feature vectors, and the element values of the query vector are decided independent of whether the keywords are included in all elements in the partial documents. Therefore, we can express the query feature vector $\mathbf{q}$ as follows:

$$\mathbf{q} = (q_{t_1}, q_{t_2}, \dots, q_{t_n}) \quad (2)$$

where

$$q_{t_k} = \begin{cases} 1 & \text{when } t_k \text{ is included in query keywords} \\ 0 & \text{otherwise} \end{cases}$$

Then our system calculates the similarities between the query vector and the element feature vectors. In the vector space model, the similarity score is calculated by the cosine of the angle between these two vectors as follows:

$$sim(\mathbf{q}, \mathbf{F}(e_j)) = \frac{\mathbf{q} \cdot \mathbf{F}(e_j)}{|\mathbf{q}| |\mathbf{F}(e_j)|} \quad (3)$$

Finally, our system calculates the scores of CSs and sorts these partial documents in order of their scores. The score of a CS is calculated as the average of scores of elements that are self-or-descendant of the root of the CS. In a preliminary experiment, we found that this calculation method yields better result than that which calculates similarities between the query vector and the CS's feature vector itself.

5.2 Reference Collection

When we evaluate IR systems, we usually use reference collections. The reference collection quite often used is the TREC (Text REtrieval Conference) collection.

Many kinds of reference collections have been created by NIST (the National Institute of Standards and Technology), though NIST has not yet exhibited a reference collection for XML documents. Therefore, we used XML documents obtained from the Web pages of “Technical Reports and Publications” at the W3C Web site⁴ as our own reference collection. Our reference collection has 17 translated XML documents with 4,344 element type nodes and three query/answer sets⁵. We prepared the following three query/answer sets in our reference collection:

– **query/answer 1**

question: How can the problem of the compatibility that the present XHTML is holding be solved in the future?

input keywords: XHTML, compatible, issue, future

answer: section 5 and 6 in REC-xhtml11-20000126.xml

– **query/answer 2**

question: What can the character code of the entity of XML use?

input keywords: XML, entity, character, encode, UTF-8

answer: section 2.2, 4.3.3, and appendix F in REC-xml-19980210.xml and section 2.2, 4.3.3, and appendix F.1 in REC-xml-20001006.xml

– **query/answer 3**

question: What character do you use for element type name and attribute name?

input keywords: attribute, element, type name, character, code

answer: section 2, 3, 4, and appendix A.3 in REC-xml-names-19990114.xml and section 2.2, 2.3, 3.0⁶, 3.1, 3.2, 3.3 and appendix B in REC-xml-19980210.xml and REC-xml-20001006.xml

5.3 Preliminary Experimental Results

Using our reference collection, we have evaluated the three approaches, *context-node*, *selected-node* and *all-node*, and drawn the recall-precision graph⁷. In the *selected-node* approach, we have used the partial documents showing chapter, section, or subsection. From the nature of the approaches, the number of CSs in the *context-node* or *selected-node* approach is less than that in the *all-node* approach. In our reference collection, the number of CSs extracted is 1,191, 1,093 and 4,343, in the *context-node*, *selected-node* and *all-node* approaches, respectively. The time taken to extract CSs depends on the performance of XML

⁴ <http://www.w3c.org/TR/>

⁵ This reference collection was made by our laboratory members. We investigated 17 documents and generated three query/answer sets

⁶ This section is some paragraphs located between captions of section 3 and section 3.1.

⁷ We adopt the drawing method of precision versus recall curve as the interpolated precision at 11 standard recall levels. This method which is usually used for illustration of the precision versus recall curve is described in [3].

parser. In these experiments, we were able to extract the CSs and generate an index within one minute.

Fig. 5 shows the average recall-precision for the three set of input keywords for the query/answer sets described in section 5.2. In a comparison of these

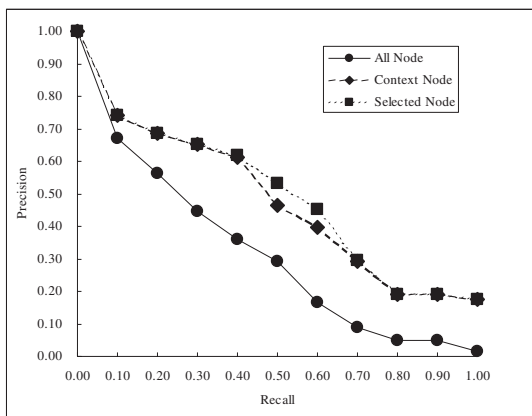


Fig. 5. Recall-Precision Graph.

graphs, we found that the *context-node* approach can retrieve more relevant partial documents than the *all-node* approach can. In fact, because some partial documents generated using all of the intermediate nodes include some irrelevant elements, we found that the *all-node* approach is not suitable in this experiment. The *all-node* approach is not bad approach, however, the number of partial documents extracted from same reference collection is extremely large, so that it takes much time to obtain retrieval results. Therefore, we think the *all-node* approach is suitable for XML IR systems. Moreover, the *selected-node* approach, which is used for retrieval of XML documents when DTDs are analyzed beforehand, is the most suitable retrieval method, because the system determine candidates for partial documents beforehand. However, as we discussed in section 4.2, the *selected-node* approach is not feasible in that it assume the availability of DTDs of all XML documents in the search engine, and that manual analysis of all DTDs are made by the domain analyst. By contrast, our proposed method, the *context-node* approach, can find context nodes automatically, even if the DTDs of XML documents are not available. For that reason, these two approaches need to be used properly in XML IR systems.

In summary, we have verified the effectiveness of the *context-node* approach, because it can search partial documents at a retrieval accuracy equal to the *selected-node* approach.

6 Conclusion

XML is emerging as the standard format for representing data and documents on the Internet, and a new technique to retrieve XML documents is required. We foresee that end-users' typical XML queries will be as terse as queries with current Web search engines: that is, queries will be composed of several keywords. Also, end-users will not have for knowledge about document structure. Based on the consideration, in this paper, we proposed a retrieval method of context search, that is an approach to automatically identifying coherent partial documents without the knowledge of XML document structures. This method integrates structure based and IR-based approaches for XML documents. Using our proposed method, we can retrieve relevant partial documents with the retrieval accuracy as same as using *selected-node* approach, even if the DTDs of XML documents are not available. Moreover, we examined the validity of our proposed method using recall-precision graph.

We address the following issues as future work:

- In this paper, we discussed a brief statement on the efficiency of *context-node* approach. Therefore, we have to carry on an in-depth experiment on our method using other large reference collection.
- Our goal is to develop a system that can support both keyword-base query and XML query language when a user inputs a query into the system. Our current system keeps the original documents and their indices for information retrieval of XML documents: that is, a large amount of disk space is needed. Therefore, a method for size reduction of both original documents and index files is needed. We consider that the BUS technique [20] will be adopted in our system.
- To develop the XML search engine, it is important for end-users to propose not only retrieval method for XML partial documents but user interface of the XML search engine. As a result, we are planning to propose user interface for end-users using similarity scores of CSs.

Acknowledgments. This work was partly supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under grants 11480088, 12680417, and 12780309.

References

1. A. Bonifati and S. Ceri. Comparative Analysis of Five XML Query Languages. *SIGMOD Record*, 29(1):68–79, 2000.
2. T. Bray, J. Paoli, C. Sperberg-McQueen, and E. Maler. Extensible Markup Language (XML) 1.0 (second edition). <http://www.w3.org/TR/REC-xml>, 2000.
3. R. Baeza-Yates and B. Ribeiro-Neto, editors. *Modern Information Retrieval*. ACM Press, 1999.
4. J. Clark and S. DeRose. XML Path Language (XPath) Version 1.0. <http://www.w3.org/TR/xpath>, 1999.

5. D. Chamberlin, J. Clark, D. Florescu, J. Robie, J. Siméon, and M. Stefanescu. XQuery: A Query Language for XML. <http://www.w3.org/TR/xquery>, 2001.
6. D. Egnor and R. Lord. Structured Information Retrieval using XML. In *Proc. of the ACM SIGIR 2000 Workshop on XML and Information Retrieval*, 2000.
7. N. Fuhr and K. Großjohann. XIRQL: An Extension of XQL for Information Retrieval. In *Proc. of the ACM SIGIR 2000 Workshop on XML and Information Retrieval*, 2000.
8. N. Fuhr and K. Großjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 172–180. ACM Press, 2001.
9. D. Florescu, D. Kossmann, and I. Manolescu. Integrating Keyword Search into XML Query Processing. In *Proc. of the 9th International World Wide Web Conference*, 2000.
10. R. Goldman and J. Widom. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. In *Proc. of 23rd International Conference on Very Large Data Bases*, pages 436–445. Morgan Kaufmann, 1997.
11. R. Goldman and J. Widom. Approximate DataGuides. In *Proc. of the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats*, 1999.
12. B. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227, 2000.
13. H. Katz. XML Query Engine. <http://www.fatdog.com/>, 1999.
14. H. Kinutani, M. Yoshikawa, and S. Uemura. Identifying Result Subdocuments of XML Search Conditions. In *Proc. of the 2000 Kyoto International Conference on Digital Libraries: Research and Practice*, pages 232–239, 2000.
15. A. Le Hors, P. Le Hégarret, L. Wood, G. Nical, J. Robie, M. Champion, and S. Byrne. Document Object Model (DOM) Level 2 Core Specification Version 1.0. <http://www.w3.org/TR/DOM-Level-2-Core>, 2000.
16. G. Navarro and R. Baeza-Yates. Proximal Nodes: A Model to Query Document Databases by Content and Structure. *ACM Transactions on Information Systems*, 15(4):400–435, 1997.
17. J. Robie. XQL (XML Query Language). <http://metalab.unc.edu/xql/xql-proposal.xml>, 1999.
18. G. Salton. *Automatic Information Organizations and Retrieval*. McGraw-Hill, 1968.
19. G. Salton and C. Buckley. Term-weighting approaches in automatic retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
20. D. Shin, H. Jang, and H. Jin. BUS: An Effective Indexing and Retrieval Scheme in Structured Document. In *Proc. of the 3rd ACM Conference on Digital Libraries*, pages 235–243. ACM, 1998.
21. A. Theobald and G. Weikum. Adding Relevance to XML. In *Proc. of the Third International Workshop on the Web and Databases*, 2000.

XDoC-WFMS: A Framework for Document Centric Workflow Management System

Rupa Krishnan, Lalitha Munaga, and Kamalakara Karlapalem

International Institute of Information Technology,

Gachibowli, Hyderabad - 500019, INDIA

www.iiit.net; {rupa,lalitha}@gdit.iiit.net; kamal@iiit.net

Abstract. Document Management is an important component of a workflow management system. XML has become a prominent language for document processing and management, and has an additional advantage of managing persistent documents using XML document management system. In this paper, we develop an XML Document Centric Workflow Management System (XDoC-WFMS) for specifying, executing and controlling workflows. A workflow consists of a set of interrelated activities. These activities involve accessing multiple documents, each of which is encapsulated within a micro-agent. A micro-agent is a system agent within an XDoC-WFMS that performs document management (i.e., retrieving, concurrent updating, and transferring documents). The XDoC-WFMS supports seamless integration of document processing within a workflow, flexible and dynamic management of XML documents, micro-agent based event and exception handling, and secure document access.

1. Introduction

Information flow through an organization can be both complex and intricate. This sophistication arises from the specific rules and constraints within business processes inherent in each organization. There are some processes that are wholly centered on document movements. Some common examples of this are Newspaper Editing, Graduate Admissions, Job Application Processing and tendering and procurement. Document flow through an organization can be modeled in terms of workflows. With the advent of XML, most of the organizations are adopting XML as the document description language. Documents in XML have certain inherent structure, which can be exploited to have a document flow specification. Documents have sub-documents that can be collaboratively worked on by different users. Any document processing requires routing of documents, checking the integrity of the documents, retrieving and combining various (sub-) documents for an activity to start executing, and finally, once the activity completes execution, handling the (sub-)documents and routing them to other activities.

A workflow in a business organization is more than just document processing, it involves other activities, such as, authorization/approval, executing certain activities and communicating certain decisions. Therefore, there is a need to have a seamless integration between document processing and other activities being executed. A document centric view of a workflow management system that caters to other activities facilitates this integration. In this paper, we address this problem by proposing an XML document centric workflow management system (XDoC-WFMS),

which exploits the organization of the XML documents while having the full functionality of workflow management system to execute other activities. Notions of views, templates, document read units, smallest concurrency unit of a document are used along with micro-agents to provide support for document handling in a user transparent manner.

1.1 Contributions and Organization of the Paper

The main contributions of the XML Document Centric Workflow Management System:

- Document Centric Work flow modeling, which allows for distinct document flows to be embedded within workflows.
- View based document interfacing, which allows the user to access only those (sub-)documents (s)he is restricted to view. This allows for the implementation of least privilege security mechanisms.
- Micro-agents are used to handle document transfers and entries into the database. Micro-agents are also used to handle some of the exception conditions implicitly within the XDoC-WFMS.
- A workflow management system that facilitates execution of document-processing activities and other activities in a seamless manner. A framework for full functionality of a workflow management system based on capability modeling and event-driven execution is presented.

The rest of the paper is organized as follows, Section 2 describes the architectural model of XDoC-WFMS, Section 3 describes concepts and terminology and the specification, Section 4 elaborates an Example, Section 5 describes the Execution Model of XDoC-WFMS, Section 6 presents the related work, and Section 7 the conclusion.

2. Concepts, Terminology, and Specification

A *Workflow* aims to automate business processes, where documents and information are passed between agents according to a set of rules to achieve or contribute to an overall business goal. A *Workflow Management System* is the software for specifying, executing and monitoring workflows. Modeling a business process would involve describing *users* who interact with the workflow management system to perform *activities*, *documents* that the users modify and share, and the internal structure of the workflow. The concepts used, and introduced in XDoC-WFMS are described below.

Template : A template abstracts logically related data of the workflow. The execution of an instance of the workflow is equivalent to populating the template instances of the workflow. A template is an XML document while the read units and modifiable units within a template are XML sub-documents.

Templates are specified as classes in C++. Table 1 shows the C++ class specification of a template. Template objects are created for each instance of workflow execution. Every template has an *identifier* and a *description*. It also contains a recursive structure representing the elements in the template, the element structure starts with the root node of the document and its child elements are all the child nodes of the

root. The child elements are specified in the Element Structure by giving the root of the child element.

Table 1. Template, View and Agent Class Specifications

Class template { struct Child_Element { Element_Name; Attribute[]; Child_Element[]; } Template_Id; Template_Description; DTD { DTD rules } Child_Element root; }	Class view { viewid; viewname; agentrolename; Struct Read_Element { name; SCUId; elementdata; } Element_Structure //ES { SCUs[]; Read_Element Read_Elements[]; } }	Class agent { agentrole; agentid ; name; capabilities[]; Activities[]; }

Table 2. SCU Class and Object Specification

class SCU { SCU_Id, SCU_Name, Element_Structure//ES { Element_Name Template_Id ES Child[]; } }		
	SCUId	1
	SCUname	StudentInfo
	Element_Structure	
	ElementName	StudentInfo
	TemplateId	1
	Element_Structure Child[1]	
	ElementName	Address
	Element_Structure Child[1]	NULL
	Element_Structure Child[2]	
	ElementName	Address
	Element_Structure Child[1]	NULL
	Element_Structure Child[3]	
	ElementName	Qualification
	Element_Structure Child[1]	NULL

Read Unit: Read Units are non-modifiable information units. These units are defined across elements of one or more templates, and reflect the data stored in the elements, in instances of the templates at time of user access. These units may differ from one user to another. The Read Unit is specified as class, containing element_ids and corresponding template_ids.

Smallest Concurrency Unit (SCU): An SCU is the atomic updatable unit of a document. Simultaneous update access by multiple users/agents to an SCU is not permitted. However, simultaneous update of template data by different users/agents accessing different SCUs is possible therefore facilitating parallel execution of

workflow activities and increasing the throughput. The data updated in the SCU is stored in the template instance. Each SCU contains a set of modifiable elements. An SCU can either be defined over a single template or across different templates. The class specification of an SCU is given in Table 2. In this specification, the element structure contains the element name, which is the name of the root element, and a field for template id. This field is filled only if the whole root node is taken from one template, else the field is set to null. The children are again nodes of type Element Structure, and they can be taken from some template, or specified by the WFMS-administrator. The SCU object, shown in Table 2, is instantiated at the time of View Creation.

View : A View is a collection of SCUs and RUs seen by an External Agent/user (EA) (described later). A view is generated dynamically from its base templates during workflow execution. For each of the Agent Roles in the XDoC-WFMS, a View is defined. Views can also be defined for individual EAs, but by default the view of the EA is the view of its Role. The EAs view the workflow system through these Views. Each EA is permitted to access to only a part of the system. Therefore, each EA is provided with a View that consists of elements which (s)he needs for performing an activity and has permission to. A view can contain one or more SCUs. In addition to the elements of SCUs, a view can also have other elements that are used only for information/display purposes. These constitute the read units of the view.

Table 1 gives the specification that helps the WFMS-administrator to describe a view. In this specification, *viewid* uniquely identifies the view, *viewname* is any name given to the view by the WFMS-administrator, *agentrolename* specifies to which agentrole this particular one serves as a view. The *SCUs* array specifies the different SCUs that belong to this view. *Read_Elements* array specifies the details of RUs in the View.

Workflow: A Workflow describes the business process. It represents the final business goal that has to be attained. It has a start state, an end state and some error and exception states. The start state represents the beginning of the workflow, the end state represents the attainment of the goal, and error states indicate exceptions. The workflow decomposer, shown in Fig 1, decomposes the workflow specified by the WFMS-administrator into activities.

Sub-workflow: Sub-workflows represent sub-goals of the workflow. Their interaction is modeled using ECA rules [2,3]. A sub-workflow can be further subdivided into activities. An Activity is the atomic unit of work, and the interaction among sub-workflows is represented as follows. The concept of sub-workflows only helps in facilitating the specification of the workflow, while the execution is modeled at the activity level.

1. An activity is a (sub)-workflow S_i .
2. A conjunction of sub-workflows is a workflow. $W = S_1 \wedge S_2 \wedge \dots \wedge S_n$, where, W represents the Workflow and $S_1, S_2, \dots, S_n$ the sub-workflows. Here the workflow is said to be executed only if all the sub-workflows are executed.

3. A disjunction of its sub-workflows is a workflow. $W = S_1 \vee S_2 \vee \dots \vee S_n$ the workflow W is executed only if one or more of S_i 's execute successfully.
4. A sequence of sub-workflows is a workflow. $W = \text{Seq}(S_1, S_2, \dots, S_n)$ the workflow W is executed only if sub-workflows execute successfully in the following sequence $S_1, S_2, \dots, S_n$.
5. A finite number of applications of above set of rules on a set of sub-workflows is also a workflow. For example, $W = \text{Seq}(S_1, (S_2 \wedge S_3), (S_4 \vee S_5))$

The workflow administrator decomposes workflows into sub-workflows, and sub-workflows into activities, using the decomposition algorithm [1].

Activity: An Activity is an atomic unit of work and is performed by a single agent. An activity involves:

1. Interacting with Human Users through Views
2. Interacting with External software or hardware systems.

External Agent: An External Agent (EA) is a human, hardware or a software agent that is the recipient of some information in the workflow. The External agent interacts with the workflow by performing agent-specific functionality that contributes to the goal of the workflow. The functionalities of the EA are domain specific and are specified by the administrator. An EA's interaction with the workflow is modeled in the form of an activity.

An agent is capable of doing certain activities in the workflow. This agent is then said to possess *capabilities* [1] for performing the above activities. Therefore, EA's capabilities are used in assigning activities to it. The different EAs in the system and their characteristics are specified by using C++ classes. An EA is capable of doing many activities and an activity can be done by several EAs. Table 1 gives the class specification for an External Agent.

Agent Role: An Agent Role is an abstraction that is used to specify the basic set of permissions and capabilities for a set of similarly capable Agents.

Micro Agents: Micro Agents are internal agents of the XDoC-WFMS, which perform system specific tasks of the workflow management system. The micro-agents are semi-autonomous, reactive, pro-active entities having social ability [13].

A workflow execution involves process specific and process independent activities. Process specific activities depend on the business process and the method used for modeling it. Process independent activities, are specific to the workflow management system and define its working. Some examples of the process independent activities are data-transfer, event handling, database management, view management, exception handling and interfacing with external devices. Micro-agents perform the process independent activities of the WFMS. Some micro-agent functionalities are:

- Presenting the current data corresponding to the element from the database when the external agent initially accesses the view.
- Micro-agents support event handling at activity level.
- A micro-agent can access multiple databases to retrieve and input relevant information to the XML view being processed. For example, in the Post Graduate Admission Workflow, after the application has passed the completeness check, depending on the stream chosen by the student, a micro-agent could choose the faculty member that is capable of reviewing that particular student.

- Exception handling through user notification, user input, and automatic processing.

The properties of the micro-agents specified above support finer control of activity execution and speedier action through faster execution of activities of a workflow in XDoC-WFMS.

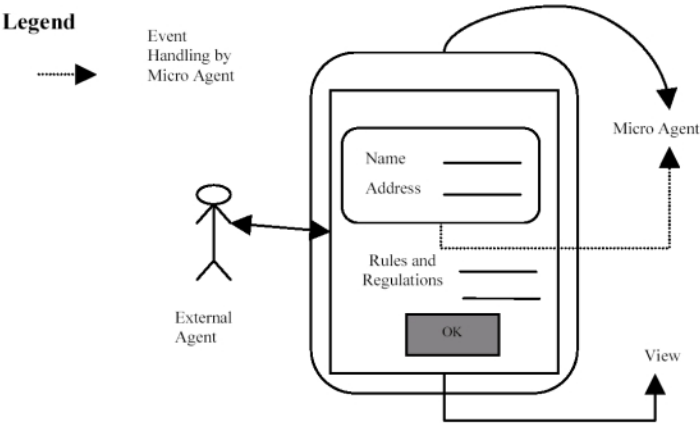


Fig. 1. Illustration of role of micro-agents in ExternalAgent-View Interaction.

Fig 2 depicts the interaction EA interaction with the view and the role of the micro-agent in it. A view is encapsulated in a micro-agent. The micro-agent handles the events generated while the view is being accessed. For example, if the RUs are updated by some other EA when this EA is viewing, the update will be notified to the current EA by the micro-agent of this view.

Administrator: The administrator of the workflow is a special External Agent, with pre-defined capabilities. The administrator plays two important roles

1. The administrator converts the business model, into a process to be executed by the WFMS, by specifying various workflow elements. (S)he specifies the Templates, SCUs, RUs, Views, Sub Workflows, Activities, ECA Rules and Agent Capabilities for a particular workflow.
2. The consistency of the above information is checked by the WFMS. The administrator monitors workflow execution to improve system efficiency and to take action in case of failures. Administrator deals with configuration management of the WFMS and failure recovery, which are outside the scope of this paper.

3. Architecture of XDoC-WFMS

The different modules of XDoC-WFMS system include Document Specification Language Processor, Workflow Decomposer, View Specification Processor, Workflow Generator, Micro-Agent Manager, Workflow Coordinator, Activity Log Manager and Recovery Manager. Fig.1 illustrates the architecture of the XDoC-WFMS. The key modules of the architecture and their roles are explained below:

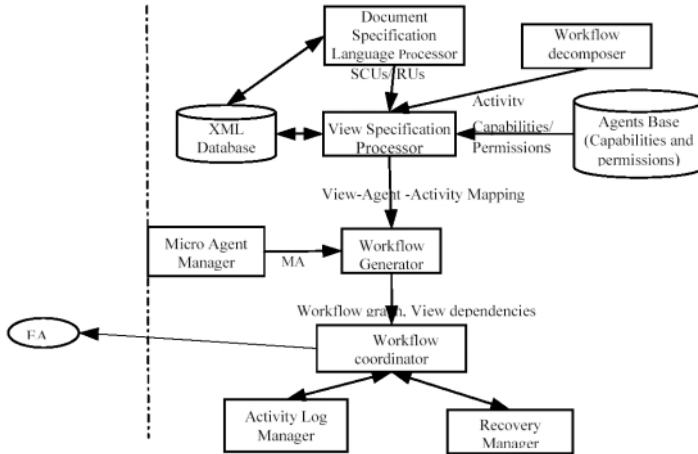


Fig. 2. Architecture of X-DoC WFMS

1. **Document Specification Language Processor:** This module allows the WFMS-administrator to specify the templates, SCUs and RUs of the workflow. The elements of the Template are stored in the XML database.
2. **Workflow Decomposer:** The workflow decomposer interacts with the administrator to decompose the workflow into sub-workflows and activities. This is done using the methodology of CapBasED-AMS [1] of decomposing workflows.
3. **View Specification Processor:** This module maps the views with their corresponding SCUs and RUs. This module also associates the Activities with the views, and the External Agents with the Activities that they can perform. It outputs a View-Agent- Activity Mapping. In case of Activities without corresponding views (i.e., Activities having control functionalities), the View Specification Processor outputs only the Agent-Activity Mapping.
4. **Workflow Generator:** The workflow generator takes a set of View-Agent-Activity and/or Agent-Activity Mappings and outputs a workflow graph. It interacts with the Micro-Agent Generator, which provides the micro-agents to handle view related activities and exception handling. This module also determines the order in which the views are accessed and therefore outputs the workflow graph. The graph represents the order in which the views are activated.
5. **Micro-Agent Manager:** With the workflow specification as the input, this module generates different micro-agents required by the workflow and associates them with the views. Apart from generating the micro-agents, this module is responsible for the various functionalities provided by the micro-agents. The different functions provided by the micro-agents are dealt with in detail in section 4.4.
6. **Workflow coordinator** The workflow coordinator takes the input as the workflow graph and executes the workflow at run-time. The objects for templates, SCUs, views, etc. are handled by the Run-time object environment.

7. Activity Log Manager/ Recovery Manager In case of run-time exceptions, control is handed over to the activity log manager and recovery manager. The micro-agents interact with them to restore the system to a consistent state.

4. Example

The working example that is used for XDoC-WFMS, is the Graduate Admission Process in [1]. The graduate admission process starts with a student filling up an admission form. This admission form is sent to the Graduate Admission Coordinator who performs the completeness check. If the document is complete, it is sent to the faculty members for evaluation. Each faculty member puts forth his comments and decisions regarding the candidate. The Graduate Admission Coordinator then decides to accept, reject or interview the candidate. The student is notified of the decision. A candidate who is called for the interview is rejected or accepted based on the interview. There are time constraints in the system. If the student admission form is incomplete, he/she has to fill it and resubmit before the last date for application forms. Similarly the interview call notification has to be acknowledged before the last date. Also the faculty members who are evaluating the applications must do so before a particular date. In case of default, the student's application is rejected and his application is removed. The activity diagram of the above example is shown in Fig. 3, along with the SCUs, RUs, Micro-agents, and Views involved in it.

There are two different templates in the Graduate Admission workflow: Student Template (has both SCUs and RUs) and Notification Template (has only RUs). An additional read-only template could be a Course template whose elements represent information about the list of courses offered, and the course details are read units in the student template. Any element of the course template is not modified in the process of student admission. Whenever a student applies for admission, an instance of the student template is created. This template contains all the information about the student admission at any point in the workflow execution. The different SCUs involved in this workflow, the various elements each SCU contains and the EAs who are given write access to these SCUs are shown in Table 3. Also the different RUs involved in the workflow are shown in Table 3. (i,j,p) in the Fig 3 represents a read unit with id j , which is a part of an SCU with id i and is given a permission p . When a student fills up an instance of the student template, each of the SCUs in this template follows a different path along the workflow execution. After the student fills in the SCU with id 1 (Name, phone, address, qualification, recommendation letters), it is sent to the Admission Coordinator as read unit 3 in Table 3. The micro-agent associated with the CompletenessCheck activity of Admission Coordinator (MA 1 in Fig 3) does this data transfer. The Admission Coordinator who has a sound knowledge of the admission procedure (KAP[1]) and also has the ability to approve student admission (CTA[1]) checks if all the details are furnished by the applicant. If a student application is complete, the Coordinator, who has knowledge of the faculty research areas (KFA[1]) sends the student details: Name, qualifications and recommendation letters (RU id 2 of Table 3) to two faculty members through a micro-agent. After the deadline of the reviews, the Coordinator views the faculty comments and decision about the student under consideration, (SCUs 2 and 3 in Table 3) and

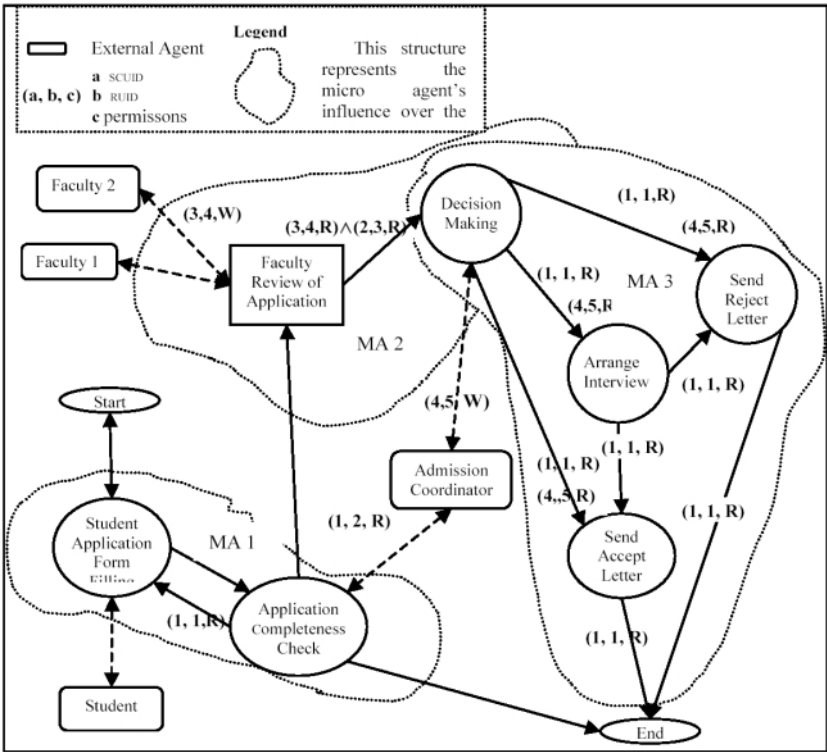


Fig. 3. Post Graduate Admission Workflow showing all the activities involved, and the agents which perform these activities.

Table 3. SCUs and RUs involved in Post Graduate Admission Workflow

Student Template SCUs			Units involved in document flow			
SCU ID	Elements in the SCU	EAs with write access	SCU name	SCU id	Read Units of this SCU	RU id
1	Name, phone, address, qualification, recommendation letters.	Student	Student	1	Name, address, phone	1
2	Comments & decision of Faculty1	Faculty1	Student	1	Name, qualifications, recommendation letters.	2
3	Comments & decision of Faculty2	Faculty2	Student	1	Name, address, phone, qualifications	3
4	Coordinator decision	Admission Coordinator	Faculty1	2	Comments & decision	4
			Faculty2	3	Comments & decision	5
			Coordinator Decision	4	Coordinator Decision	6

gives his final decision of approving or rejecting the student by modifying the *Coordinator Decision field* (SCU 4 in Table 3). The Admission Coordinator can call a student for an interview to decide on selecting or rejecting the student. Once the decision is taken, the student information (RU id 1) and the Coordinator decision, accept/reject (RU id 6) need to be sent to the student. These Read Units are passed as parameters, to the micro-agent associated with the notification activity. The notification micro-agent treats these as read units and instantiates a notification

template with the values from the read units 1 and 6. The notification micro-agent interfaces with the e-mail client, is responsible for notifying the students appropriately. Even in case of an interview, the interview notification is similarly sent to the student and after the interview is conducted, the final decision of accept/reject is taken and the student is appropriately notified. If the decision is to admit the student, the student's information is entered into the table of admitted students; else, the student record is removed.

5 Execution Model

5.1 Architecture of Workflow Coordinator

The execution model describes how a workflow instance gets executed, and describes the steps in the workflow instance life cycle. The workflow coordinator handles the workflow execution. The architecture of workflow coordinator is given in Fig. 4. The components of the Workflow Coordinator are:

- **Workflow Engine:** The workflow engine is the key component of the workflow coordinator. Its functionalities are
 - It incorporates an Event Capture Module.
 - It invokes micro agents.
 - It performs ECA Rule look up and initiates the corresponding action.
- **Micro Agent:** A micro agent performs the following kinds of functions
 - It interfaces with external software or hardware systems.
 - It creates and handles views.
 - It manages user worklists.

The micro agent lifecycle is from creation of the micro agent at start of an activity execution to the completion of the activity.

- **The ECA Rule Base and Run Time Knowledge Base:** The ECA Rule base contains the ECA rules specific to a workflow. The Run Time Knowledge Base maintains instances of various workflow elements like Templates, SCUs, Views, Activities etc. associated with the different workflow instances executing at that point of time.

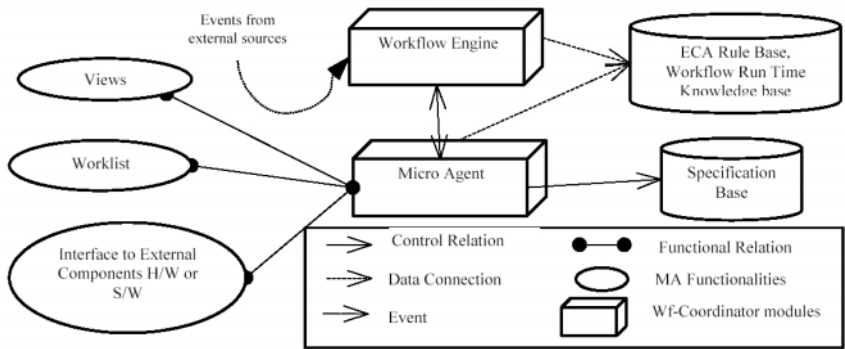


Fig. 4. Architecture of the Workflow Coordinator Module

5.2 The Workflow Execution Algorithm

Table 4. Pseudo Code for Execution Algorithm.

Pseudo Code for the Execution Algorithm 1. EventSet Ei 2. Capture_Start_Event(Ei) 3. Assign_Execution_Instance_Id(instance_id) 4. Action Ai = LookUpInECARule(Ei) 5. OutputEventSet Oi = PerformAction(Ai, instance_id) 6. Oi supersetof InputEvenSetForActivity() 7. μ = GenerateMicroAgentForActivity(instance_id) 8. Ei = PerformActivity(μ); 9. Goto Step iv
Pseudo Code of Perform Activity 1. GetActivityDescription 2. if InterfaceRelatedActivity 3. ExecuteLibraryRoutine 4. if UserRelatedActivity 5. CreateUserView 6. if WorklistRelatedActivity 7. UpdateUserWorklist

Example: The execution of the example given in Section 4 can be modeled in terms of the Execution Algorithm.

1. The student logs on to the WFMS. This generates the workflow start event.
2. The workflow engine creates a workflow instance and assigns it an id.
3. The workflow engine invokes and assigns a micro agent to the ‘Student Form Filling’ activity. This micro agent handles the Student Form Filling view events. After the successful completion of the activity, the micro agent updates the student template and notifies the workflow engine.
4. The workflow engine instantiates a micro agent for ‘Admission Completeness Check Activity’.
5. This micro agent handles the events associated with the admission coordinator view. The completion of this activity generates the incomplete_application or complete_application event.
6. The wf engine performs an ECA Rule lookup for the above events
 - The incomplete_application event is the input event to the ‘Incomplete Application Notification’ Activity. This activity involves interfacing with the e-mail client and sending the notification message as e-mail to the student. The interfacing is facilitated by library routines corresponding to that particular e-mail client. The student template information is retained till the timeout on receiving application occurs.
 - The complete_application event is the input event for faculty review activity. Each of the two faculty members evaluating the student application performs an instance of this activity. Therefore, a micro agent is invoked for each faculty member involved in the activity. These agents are invoked simultaneously.

After a faculty member finishes reviewing, the micro agent updates the student template and notifies the workflow engine about the activity completion.

7. The workflow engine invokes the next activity – ‘Admission Coordinator Decision’ only after both the faculty reviews are available. It creates a micro agent associated with this activity, which in turn handles the events associated with the Decision view. The output event of this activity is ‘Accepted’, ‘Rejected’ or ‘Interview’.
8. The workflow engine performs an ECA Rule lookup for the above events. All the three events involve sending notifications. The micro-agent interfaces with the e-mail client as described in step 6. The corresponding action for the ‘Reject’ event would also involve removing the student template from the knowledge base.

5.3 Discussion

XDoC-WFMS, was an attempt, to develop a document centric workflow management system that makes use of the inherent structural semantics of XML, and the flexibility of the agents. The concepts of SCUs, Templates, Views and micro-agents have been introduced. We have used XML as our document description language, because XML facilitates the division of documents into semantically well-structured sub documents. Templates and Views, are comparable to base tables and relational views in the relational model. SCUs were introduced to define sub-documents that facilitate concurrent access to the templates. Views were introduced to provide secure access to the system. Apart from this, they also allow for customization of user preferences. Workflows have embedded document flows in them, but this usually deals with document level movement, or message passing. But XML also facilitates sub-document flows and a refresh capability to view the current state of the database. The execution of a WFMS requires processing entities that maintain the connectivity of activities in workflow. These entities form an inherent part of the system, while ensuring consistency, and efficiency in execution. Menial WFMS tasks can be entrusted to micro agents as user performed activities tend to be slow. To increase the speed of execution of these activities, micro agents can be used to perform some amount of pre-processing and post-processing pertaining to an activity. And micro agents can be customized by changing their execution parameters. Further details about role of micro agents in XDoC-WFMS are currently being worked on.

6. Related Work

The CapBasED-AMS [1] has an architecture built around task management and user interaction with tasks. The tasks there are general and also include system specific tasks. System specific tasks are modeled as interactions with hardware or software systems and are specified by the administrator. The micro-agents in XDoC-WFMS handle this system specific activities and interface with the Views provided to the External Agents, who are the users, either human, software or hardware interacting with the system. OASIS system [5] models their WFMS around MOAPs (Micro Organizational Activity Processor). The OASIS WFMS is a network of communicating MOAPs, which are associated with knowledge bases at various levels - user, group and organizational. A task in OASIS, encapsulates both data and

processing objects. XDoC-WFMS totally separates data and processing units. The processing units are micro-agents, and data is specified in terms of Templates. This allows the administrator to specify document flows separately, without having to specify micro-agents. In [4] on secure workflows, the concept of Least Privilege

Security is dealt with. Least Privilege Security gives a user minimum access to tasks and information in the organization; the privileges are withdrawn from the user once the task is completed. This is encapsulated in XDoC-WFMS by the inclusion of

Views. The access to these views is restricted. Only an External Agent who has the right privileges can access the view and his permissions are withdrawn once the activity is performed.

The MARIFlow system [6] describes the architectural details required for automating and monitoring the flow of control and documents over the Internet among different organizations. This is achieved by using MARCAs (or MARIFlow Cooperating Agents). The MARCAs are automatically initialized at each of the sites involved in the workflow and they provide for coordination with the other agents in the system by routing the documents in electronic form according to the process description. The security constraints like organizational firewall are also handled by these MARCAs. The notion of orthogonal and yet related handling of document flows, activity control flows and WFMS functionality in a seamless integrated manner by using templates, views and micro-agents is novel and contributing aspect of this work.

Event handling during a workflow execution can be made easy by integrating event handling system into the database. This can be done by using Database Alerting Techniques [8]. This work deals with an office automation model using active databases. The updates made to the office database can trigger events based on user defined rules. For example, a rule can be specified to make an Alerter Module, monitor a particular type of an Update Operation. The Activity Management System in this model consists of an Alerting subsystem, Office manager and Activity agents. The alerting subsystem screens database updates to detect if any of the rules have satisfied so that events can be triggered. The office manager coordinates activities and the activity agent is an office specialist capable of performing some defined office activity. The user interacts through a user interface called *Intelligent Coupler* which sends appropriate messages to various modules of the AMS. In our work, we introduce the concept of micro-agents which serve this purpose.

The system OA-DBMS (Office Automation DBMS) [9] emphasizes on an integrated database support, such that data can be shared among applications. It supports the use of four data modeling concepts: aggregate hierarchies (a document is a set of hierarchy related entities which are aggregations of subtypes), generalization (abstraction of common properties of different entities), cross references (expresses subordinate relationships of entities of similar or different types) and long fields (Unstructured and multimedia data are stored as long fields).

METUFlow system[10] supports the idea of distributed execution environment. In this distributed environment, the scheduler, the history manager and worklist manager are fully distributed, giving rise to failure resiliency and increased performance. The workflow process here is defined as a collection of blocks, tasks and sub-processes and each of them is referred to as an activity.

The WASA Approach [11] deals with a flexible and platform-independent workflow support for scientific applications such as geo-processing, molecular biology and laboratory environments. In these domains, there is a need for dynamic modifications of the workflow model, while the workflow is being executed. These changes are categorized as anticipated dynamic changes and ad-hoc dynamic changes. The WASA model explains how the two types of changes are handled.

Enterprise-Wide Workflow Management Based on State and Activity Charts [12] deals with the specification, verification and distributed execution of Workflows based on State and Activity Charts. Activity Charts specify the dataflow between activities in the form of a directed graph, whereas State Charts specify the control flow between the activities and the control flow is driven by the ECA Rules. State and Activity Charts help in automatic verification of critical workflow properties by methods of model checking. Distributed execution is achieved through partitioning of the centralized workflow specification, automatically, using these graphs.

7. Conclusion

Document flow is an important aspect of workflow driven execution of business processes. But document flow and processing is not the complete workflow. Therefore, there needs to be a document centric, and yet full workflow functionality to support execution of workflows for business processes. Further, within documents there are interdependencies among sub-documents, sharing and concurrent updates to documents, and restricted access to documents. In order to facilitate the combined workflow and document flow oriented business processing, an XDoC-WFMS has been proposed in this paper.

The salient features of this system are:

- Document Centric Work flow modeling, which allows for distinct document flows to be embedded within workflows.
- View based document interfacing, which allows the user to access only those documents (s)he is restricted to view.
- Micro Agents to handle document transfers and entries into the database and also to handle exception conditions that arise during the workflow execution.
- A workflow management system that facilitates execution of document-processing activities and other activities in a seamless manner. A full functionality of a workflow management system based on capability modeling and event-driven execution is provided for this.

In our on-going work, we are implementing the XDoC-WFMS and are conducting case-studies to automate the administrative business processes in the institute.

References

- [1] K. Karlapalem, H. P. Yeung, P. C.K.Hung: "CapBasED-AMS- A Framework for Capability-Based and Event-Driven Activity Management System", CoopIS 1995: pages 205-219.
- [2] S. Chakravarthy, V. Krishnaprasad, Z. Tamizuddin, R. H. Badani: ECA Rule Integration into an OODBMS: Architecture and Implementation, International Conference on Data Engineering 1995: 341-348

- [3] S. Chakravarthy, R. Lee, R. Dasari: "ECA-Rule Processing in Distributed and Heterogeneous Environments", *Distributed Objects and Applications* 1999: 330-339.
- [4] P. C. K. Hung : "Secure Workflow Management Systems", Ph.D. thesis, Department of Computer Science, The Hong Kong University of Science and Technology, July 2001 .
- [5] C. Martens, F. H. Lochovsky: "OASIS: A Programming Environment for Implementing Distributed Organizational Support Systems" , Conference proceedings on Organizational computing systems, 1991, *Pages 29 - 42*
- [6] A. Dogac, Y. Tumbag, A. Tumer, M. Ezbiderli, Nesime Tatbul, N. Hamali, C. Icdem, Catriel Beerli: "A Workflow System through Cooperating Agents for Control and Document Flow over the Internet", *CoopIS 2000* : 138-143.
- [7] K. Karlapalem, A.R. Dani, P. Radhakrishna: "A Frame Work for Modeling Electronic Contracts", to appear in *ER-2001, Japan, 2001*.
- [8] Jo-Mei Chang and Shi-Kuo Chang: "Database Alerting Techniques for Office Activities Management", *IEEE Transactions on Communications*, Vol. Com-30, No.1, Jan 1982 : Pages 74-81
- [9] M. Bever and D. Ruland: "Aggregation and generalization hierarchies in office automation", Conference Sponsored by ACM SIGOIS and IEEECS TC-OA on Office information systems, 1988 : Pages 250 - 264
- [10] A. Dogac, E. Gokkoca , S. Arpinar, P. Koksai, I. Cingil, B. Arpinar, N. Tatbul, P. Karagoz, U. Halici, M. Altinel : "Design and Implementation of a Distributed Workflow Management System: METUFlow" , *NATO ASI Series, Computer and System Sciences*, Vol.164: Pages 61-91.
- [11] G. Vossen, M. Weske: "The WASA Approach to Workflow Management for Scientific Applications", *NATO ASI Series, Computer and System Sciences*, Vol.164: Pages 145 – 164.
- [12] P. Muth, D. Wodtke, J. Weissenfels, G. Weikum, A. Kotz-Dittrich: "Enterprisewide Workflow Management Based on State and Activity Charts", *NATO ASI Series, Computer and System Sciences*, Vol.164: Pages: 281-303.
- [13] G. Joeris, C. Klauck; O. Herzog: "Dynamical and Distributed Process Management based on Agent Technology", in *Proc. of the 6 th Scandinavian Conference on Artificial Intelligence (SCAI'97)*, August '97, Helsinki, Finland, 1997: Pages 187-198.

Active XML Schemas

Michael Schrefl and Martin Bernauer

Department of Business Information Systems,
University of Linz, Austria
{schrefl | bernauer}@dke.uni-linz.ac.at

Abstract. The possibility to share content easily among e-business partners constitutes one of the Web's potentials. Especially XML facilitates interoperability between different e-business partners by providing a standard representation of content. But the support of reacting automatically to changes in related documents is still weak. Active XML Schemas extend XML Schemas in that respect.

Active XML Schemas employ concepts from active database systems, from conceptual modeling of business rules, and from event-based systems to provide for automatic and asynchronous management of distributed web content. Different to previous approaches of employing the event-condition-action (ECA) paradigm in a web-based setting, active XML Schemas integrate passive and active behavior smoothly into document schemas in an object-oriented way, treat events as "first class elements" that can be stored and queried just as other document data, and provide for logical events that are defined upon querying past and scheduled events.

1 Introduction

Sharing content is the key success factor of the Web, recently facilitated by XML [7], which supports among others the exchange of information between e-business partners. XML standards and frameworks for e-business, such as ebXML [1] or BizTalk [2] further foster the exchange by providing for interoperability.

While XML provides for a standardized textual representation of business content, easing its exchange, other problems remain unsolved. Web documents of e-business partners are often change-dependent. E.g., if the price for an overnight stay in a certain accommodation rises, travel agencies selling trips that use this accommodation might want to raise the price of these trips accordingly. Change dependencies between documents are usually characterized by autonomous control over the participating web documents, since they reside under authority of different e-business partners, and are therefore loosely coupled.

Interoperability between different applications used by different e-business partners for maintaining content can be reached by defining an XML layer on top of those applications. Thereby content residing for example in databases or workflow systems can be easily exchanged with partners. Research has been undertaken in this field, proposing several approaches (e.g., [8, 12, 13, 26, 27]), describing how XML content can be imported into and exported from databases, such as using database triggers to push changes in the database to web documents [22, 23, 25]. Propagating changes between web documents has received less attention so far. Furthermore for web-only

content, i.e., content in web documents that is not drawn from underlying databases, a light-weight web-based solution to web content management is a better alternative than forcing users to store this content in databases, too. This should not replace but complement database or workflow system functionality.

Managing change dependencies is only one aspect of web content management. Support for automatic posting of up-to-date content or unposting of outdated content, such as archiving job offers after application deadlines, is equally important. A suitable solution for the automatic web content management needs to take into account that documents are autonomous, loosely coupled storage units, whose synchronization mostly takes place asynchronously, and that changes may depend on document histories.

The concepts of active database systems are perfectly suitable for this application. Active database systems use event-condition-action (ECA) rules to define how to react to occurred events automatically. A wide range of application areas exists for ECA-rules in the field of databases, ranging from internal database usage such as constraint enforcement, or materialized view maintenance, to external applications such as monitoring states, maintaining related information, or implementing business rules. Employing active behavior in XML documents opens those application areas to XML, replacing old techniques for implementing active behavior, like immediate checking a document/application after an event occurred, or periodically polling a document/application for changes. Active behavior provides several benefits for the definition of e-services. Due to the simplicity of the active paradigm, e-services are easier to design and can evolve better than traditional approaches [5]. While [5] demonstrates the use of the active paradigm for the implementation of alerters, personalization of web pages, and for view maintenance, in [6] the active paradigm is applied to XML documents to push reactive services to XML repositories. Both approaches are based on simple mutation events occurring on XML documents as defined by the DOM level 2 Event Specification [19]. Active behavior is also utilized in [3], describing a system for building Web applications.

Active XML Schemas take the next step in that they move from simple to high-level support for active behavior within XML. They build on experience from conceptual modeling of business rules by the means of situation/activation diagrams [15]. These diagrams are characterized by modeling events as first class objects that have an event type and are collected into event classes. Beyond supporting different kinds of events (e.g., calendar events, method events, and abstract events), they allow to schedule future events and provide the functionality of composite events through logical events, which are defined declaratively by querying the extension of other event classes rather than by a special event composition language, as it is the case with active database systems [18]. Active XML Schemas reuse these concepts in the realm of XML.

Furthermore, active XML Schemas utilize the publish/subscribe model from event based systems, taking the loosely coupled nature of distributed web documents into account, to communicate events between documents. By this means, events that occur at publishing documents are delivered to subscribing documents, allowing them to react accordingly. To provide for *communication transparency*, documents export/import events to/from other documents at the conceptual level without stating whether published events are communicated to subscribers at the realization level by a push or a pull-based approach. Moreover, the event-based model allows to treat reactions to changes in other web documents or to changes in databases in a transpar-

ent way. Rules in active XML Schemas that react to events need not be aware of the source of the triggering event, thus providing *event source transparency*. To maximize design autonomy and to couple different documents as loosely as possible, all components of a rule (i.e., event, condition, action) defined for a web document are local to the document. Sharing of events and actions is only possible via communicating events to other documents.

Most important, active behavior is provided by extending XML Schemas such that active behavior comes with an XML Schema document and is used for all document instances of the schema. This is line with the current trend in web-based information systems to learn from years of research in conceptual modeling and to apply schemas for modeling and implementing web content. Such an approach provides for the reuse and interoperability of not only structure, but also of active behavior. Active behavior defined at the schema level is also easier to design, to implement, and to maintain than it would be if it were defined separately. Furthermore, the approach taken with active XML Schemas is *document centered*, meaning that events occurring at a document are stored in the document. This mimics traditional paper form processing, where all events related to a business case are recorded with its form and can be queried when decisions on the case have to be made.

The remainder of the paper is organized as follows: Section 2 presents basic extensions to XML Schemas to provide active behavior. Section 3 introduces advanced concepts such as timestamping of events, scheduling events, logical event classes, and exception rules. Section 4 concludes the paper by discussing briefly how active XML Schemas can be managed.

2 Active Extension of XML Schema

Adhering to an object-oriented approach, XML Schemas are enriched with additional components to provide for active and passive behavior of XML documents. While several schema languages have been proposed recently (e.g., [11, 14]) to replace XML 1.0 Document Type Definitions [7], we use XML Schema [4, 24].

An active XML Schema defines a distinguished global complex type, the *active document type*, that defines the structure and behavior of its instance documents. This is done by enriching a global complex type as defined by XML Schema with the definition and implementation of operations (i.e., passive behavior), with a set of definitions of event types and event classes, with a set of import and export statements defining the publication and subscription of event classes, and with a set of rules (i.e., active behavior).

To assure interoperability with legacy applications (i.e., applications that are not capable of interpreting active XML Schemas) the enrichments are transparent to those applications. Transparency is obtained by enriching complex type definitions via annotations that hold data defining active and passive behavior. XML Schema documents are instances of the XML Schema schema for XML Schemas (i.e., the metaschema [24]). To be able to validate an active XML Schema with a conventional XML Schema validator, this metaschema is enriched with metatype definitions for active and passive behavior, constituting the active XML metaschema. Like active XML Schemas are instances of the *active XML metaschema*, active XML documents are instances of active XML Schemas. They comprise, aside of static data, data repre-

senting events as well as subscriptions. These additions belong to a separate namespace, which is referenced by namespace-prefix act: in this paper.

We use an extended UML [21] notation to model active XML Schemas. It is necessary to extend UML to model concepts not originally present. We introduce new sections that a type can feature, and tag each section by a letter to reflect its semantics. While we do not use the section for structural attributes in this paper, we tag the operation section by “O”. The newly introduced event class section is tagged by “E”, the exported event class section by “↑”, the imported event class section by “↓”, and the rule section by “R”.

In the following, we describe the basic behavioral components of active XML Schemas.

2.1 Passive Behavior

So far passive behavior for XML documents is specified procedurally, i.e. separately from the data, falling behind the prevailing object-oriented paradigm that integrates structure with behavior. This is the case with DOM, SAX, or XSLT [10]. However, XML databinding can be utilized to narrow the gap between definition of data and behavior. In particular Sun’s JAXB [20] can be used to define a mapping from XML to Java classes, behavior can be added by extending these classes.

Whereas XML databinding does not allow to integrate the definition of data and behavior, active XML Schemas define operations at XML document types in an object-oriented manner. They enrich XML types, which before described only structure, with passive behavior. Interfaces and implementations of operations are separated as in common practice in object oriented design.

In this paper, implementations are illustrated by XSLT templates, which can access an operation’s parameters via variables. They are, before being applied to an active XML document, embedded into a default stylesheet that performs a carbon copy (they override the default stylesheet’s templates). The transformation’s output document replaces the original active XML document, thus performing the modification as described by the operation’s templates.

Example 1. A job agency provides an active XML Schema defining active document type `j:JobAnnounce` and a document `academicJobs.xml` having that type, which comprises a list of current job offers (cf. Figure 1). A new job offer is announced by the invocation of operation `announce(j:Job)`, which adds a new job offer at the end of the list. The part of the active XML Schema document representing document type `j:JobAnnounce` that defines the interface of operation `announce(j:Job)` is depicted in snippet 1a. The operation’s implementation with XSLT is depicted in snippet 1b. The type or object a snippet is part of, is represented by references to snippets in the extended UML model.

2.2 Event Types and Event Classes

Events are happenings of interest to a document. They are collected into event classes and are stored with documents as “first class XML elements” as any other XML ele-

ment. Each event class has a member type that defines the structure of its member events.



Fig. 1. Operations and Event Classes at Schema and Instance Level

Events can be distinguished into several *event categories*, which are: mutation events, operation events, calendar events, abstract events, and imported events. *Operation events* reflect the execution of an operation. *Mutation events* reflect changes to XML documents as according to the DOM Level 2 Event Specification. They are relevant in particular when a document is manipulated by means of legacy applications and not by predefined operations. *Calendar events* constitute past, current, or future calendar entries. *Abstract events* comprise any other event explicitly raised or scheduled by users or rules. *Imported events* wrap remote events and are discussed in the next subsection.

Event types form a type hierarchy, which is not shown for space limitations. Each event type defines various *parameters* reflecting the environmental setting in which it occurs. E.g., an operation event type will provide a parameter for each operation parameter. The root event type defines the lowest common denominator of parameters (such as an occurrence timestamp, or an identifier). An event class is either a basic event class or a logical event class. The extension of a basic event class is defined by collecting every detected event of its member type. The extension of a logical event class is defined by a query over other event classes. Logical event classes are discussed in Section 3.

Events are stored as “first class elements”. As such they are part of a document and are thus persistent. This is different to most active database systems, whose events have a life span ending with the transaction in which they occurred. The extensions of all event classes of a document reflect the document’s *event history*. Rules can query event classes and, thus, react according to the event history reflecting the document’s past as well as the documents future as determined by scheduled events.

Event classes are modeled explicitly at the schema level with document types and define the collections of events that are of interest at the instance level. When an event class is defined with a document type, each document instance of that document type will maintain its extension of the event class. *Example 2.* Figure 1 shows the definition of event class **announce** with document type **j:JobAnnounce**, collecting all **announce** operation events representing invocations of operation **announce** (cf. snippet 2), and the materialization of this event class in document instance **academicJobs.xml** (cf. snippet 3).

2.3 Import and Export of Event Classes

Not only events that occur at a document locally may be of interest, but also events that occur in another document. From a document's point of view, one can distinguish between *local events* and *remote events*. A local event happens at the document itself; a remote event happens at another, remote, document. To be able to react to remote events locally, remote event classes can be imported if they are exported by the document type of the remote document. In such a case, a local event, referred to as imported event, wraps a copy of the remote event. Replication ensures that the imported event is available independently of the remote document, which is especially of importance if the remote document is not under control of the owner of the local document, a common setting in the Web environment.

Documents will often import several event classes from the same document. Modeling import relationships between documents at the schema level can support such a setting by separating the establishment of an import relationship between document types from the definition of which event classes are imported in that relationship. This separation is achieved by defining a typed document proxy at the schema level that acts as an intermediary between the importing document type, which defines the proxy, and a document of the proxy's type. A proxy's value specifies the document event classes are imported from, it is bound at the instance level. Thus, to import an event class from another document the following steps are required: (1) at the schema level: (a) export the event class at the exporting document type, (b) define a document proxy at the importing document type, and (c) import the event class at the importing document type, and (2) at the instance level: at the importing document, bind the document proxy to the URI of a particular document,

Example 3. Figure 2 depicts an import scenario. As a courtesy to its staff and students, a faculty posts relevant job offers supplied by a job agency at its document of active document type `u:Faculty`. To provide others access to newly announced jobs via an event class, the job agency exports event class `announce` (cf. snippet 1). Active document type `u:Faculty` imports this event class, using proxy `jobSite` to refer to the document from which the remote event class is imported (cf. snippet 2). Its value is bound in instance `science.xml` to document `academicJobs.xml` (cf. snippet 4). Announced job offers are now locally available within a faculty's page in the form of events.

Often a document needs to import event classes from several documents of the same document type, which corresponds to a "1:n" relationship in conceptual modeling terms, while we considered only "1:1" relationships so far. In such a case, a *set proxy* is defined at the schema level and bound to multiple URIs at the instance level. Imported event classes of the same kind are collected into an *event class family*. When querying event class families, member qualifiers are used for selecting specific event classes. An unqualified reference to an event class family refers to the union of all events of all family members.

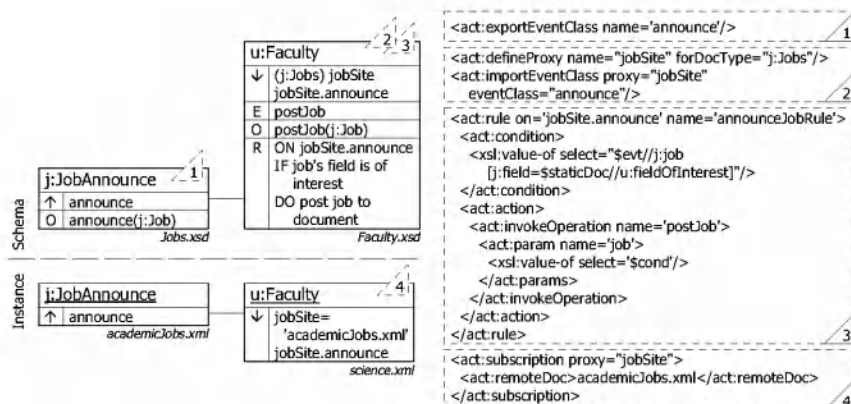


Fig. 2. Export and Import of Event Classes and Rules

Example 4. A faculty comprises several departments, each one featuring its own document of type `u:Department`. A publication is added to such a document by invoking operation `published(u:Pub)`. A faculty wishes to get notified whenever a publication is added to one of its departments' documents. Thus it imports event class family `published`, using set proxy `depts` to refer to its departments' documents, as depicted in Figure 3.

2.4 Rules

Active behavior is defined by *ECA rules* with a document type. Each rule is defined upon an event class, and consists of a condition and an action. If an event of the event class occurs, all rules defined on that event class are triggered. After a rule is triggered, its condition is evaluated, and if the condition applies, the action of the rule is executed.

To ensure loosely coupling between documents, rules are restricted to have a *local scope*, which means that event class, condition, and action may refer only to the document with whose document type the rule is defined. This corresponds to the "Law of Demeter" [17], which suggests a restricted scope for type's operations in object-oriented design and is generally considered beneficial. If a rule shall be triggered upon events of a remote event class, the event class has to be imported. If a rule shall be triggered upon rule execution in remote documents, an event must be defined upon the rule's action, exported to and imported by the other document, which can react to this imported event by a local rule. Furthermore, conditions can only access data from the local document, in order to be completely independent from other documents, and actions of rules may only invoke operations that modify data of the local document.

Conditions are queries on the static data and the event history of a document. The latter two are available to the query as bindings `staticDoc` and `evtHistory` respec-

tively. The event that triggered the rule is available as binding `evt`. Conditions apply when the query's result is not empty. We utilize XSLT for the definition of conditions because of the widespread support and knowledge of XSLT. Alternatively, XQuery can be used as well, if XQuery queries are translated to XSLT by an XQuery-to-XSLT translator [16].

Actions can invoke operations on the local document and raise or schedule events in a local event class (scheduling events is described in Section 3). As is the case for conditions, the document's data, its event history, and the triggering event are available to the action as bindings. Moreover the result of the condition evaluation is available as binding `cond`.

Rules defined at the schema level with document types may need to react specifically depending on the document instance they are triggered at. Document-specific reactions can be achieved without introducing explicit rule parameters by specifying rule conditions that query the document's data the rule is triggered at. It is, however, reasonable from a design perspective, to group elements which are employed to implicitly parameterize rules into a separate part of the document.

Example 5. Figure 2 shows the active document type for faculties, which defines rule `announceJobRule` on the imported event class `announce` (cf. snippet 3). Its condition tests whether the job's field is of interest to the specific faculty by querying elements `fieldOfInterest`, whose content is, e.g., 'CS' (computer science) and 'EE' (electrical engineering) for the science faculty `science.xml`. If the condition applies, the new job is posted to the faculty's document by invoking operation `postJob(j:Job)`.

3 Advanced Concepts

In this Section, we motivate and discuss advanced concepts of active XML Schemas: event timestamps, scheduling events, logical events, and exception rules.

3.1 Event Timestamps

When an event is added to a remote event class, it is delivered to subscribing documents. Such a communication is characterized by two timestamps, the *publication time*, which records the time when the remote document publishes the event, and the *delivery time*, which records the time the event is delivered. These timestamps may be of different importance to different applications, like the time of the post mark (corresponding to publication time) is relevant to meet the deadline of a postal vote or the time of delivery to meet an application deadline.

To support these two notions of timestamps, a local *remote event type* wraps the member type of a remote event class to provide for the recording of the timestamps. An example given later shows how the distinction between publication time and delivery time can be used in an exception rule.

3.2 Scheduling Events

Comprehensive support for web content management requires not only to handle event occurrences that were caused by modifications to the document or were explicitly raised, but also to schedule events in the future. A typical application is to schedule an event in the future to trigger removal of then outdated content. With active XML Schemas, events can be scheduled by operations or rules. Such a *scheduled event* has an occurrence time in the future.

Example 6. To remove a job offer from a faculty's document when its application deadline is over, rule `scheduleJobRemovalRule` is defined on event class `postJob`, which collects invocations of operation `postJob(j:Job)`. This rule schedules the removal of a job offer by adding an event that is scheduled to occur when the application deadline ends to event class `jobExpired`. Rule `removeJobRule` defined on event class `jobExpired` removes an expired job offer, whose identifier was bound to the event, by invoking operation `unpostJob(xs:ID)`.

3.3 Logical Events

Active database system provide operational languages for defining a *composite event* from other events [9, 18] based on their occurrence time. E.g., sequence event $C = A;B$ occurs if event B occurs after event A has occurred. As events of types A and B may occur several times, the notion of *event context* has been introduced to indicate which of these occurrences are used to detect an occurrence of C . E.g., in the “recent” context the last occurrences of A and B are used, and in the “chronical” context the i -th occurrence of A is paired with the i -th occurrence of B . Various techniques based on Petri nets, state machines, and event graphs have been introduced to detect composite events incrementally. E.g., in the example of the above sequence event, an occurrence of A would start the incremental detection of an occurrence of composite event C .

Experience in modeling business rules at the conceptual level has shown that a declarative approach for defining composite events is better suited for end users (i.e., schema designers) than an operational approach, which is more appropriate for database internal usage. Therefore, active XML Schemas provide the functionality of composite events in that logical events can be defined by querying a document's event history. Different to operational languages, which detect composite events incrementally by observing events as they occur from the past till “now”, logical events query from the perspective of “now” the past history of events and may also query future scheduled events. Furthermore, this approach allows to waive consumption policies, whose effect is sometimes hard to grasp for end users, especially when several contexts need to be combined. The query approach can also cope very easily with multiple timestamps provided by active XML Schemas, avoiding a difficult adaption of the notion of “event context” to multiple timestamps. Finally, the goal of having a “light-weight” approach in the web setting suggests to reuse already available query languages rather than providing an additional composite event detector and cluttering web documents with many unfinished event detections.

Logical events are defined by logical event classes, which specify a query over the document's event history. A logical event class can be *event preserving*, in which case it selects members from other event classes, or *event generating*, in which case its member events are newly built from other events. The specification of a logical event class identifies next to its member type a set of *terminating event classes* and *participating event classes*. A logical event occurs if a member of a terminating event class occurs and the stated query over the participating event classes is satisfied.

Example 7. A faculty publishes the active researcher status of its employed researchers at the beginning of each year, which depends on a researcher's number of publications he/she published the year before. Each publication is reflected by a **published** event that is imported by the faculty's document from its departments' documents. The active researcher status is determined by operation **publishActResStatus(u:Researcher, xs:gYear)** for a given researcher and year. Due to failures, a published event may be delivered after the active researcher status has already been determined, although the event was published in time (i.e., in the year before the current year). Logical event class **pubDeliveredLate** collects published events that should have been considered when determining a researcher's active researcher status. This is the case if a **published** event (terminating event) is delivered and a **publishActResStatus** event (participating event) exists such that (1) the **publishActResStatus** event occurs before the **published** event is delivered - notice that such a comparison is necessary, since in general a participating event class may also contain future events -, and (2) the **published** event occurred the year before and concerns a paper published by a researcher the year before, for whom the **publishActResStatus** operation has already been executed for that year (cf. Figure 3).

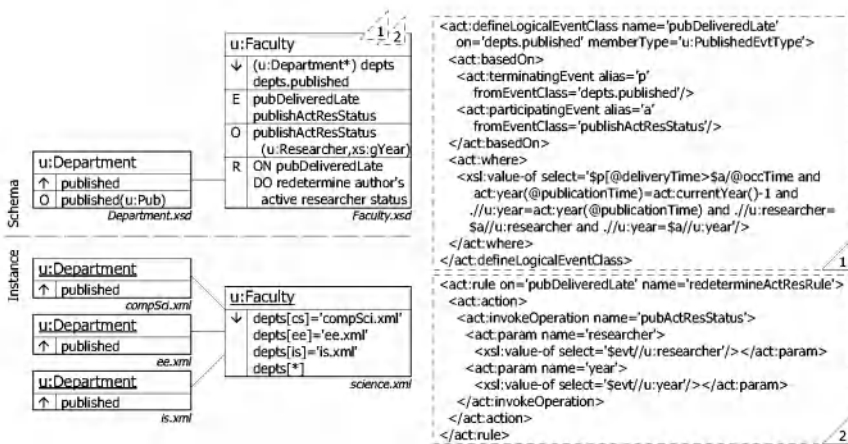


Fig. 3. Logical Event Class and Exception Rule

3.4 Exception Rules

Situations that do not constitute “standard” situations during deployment of active XML Schemas may occur. They originate in effects of an active XML Schemas’ environment. In particular extraordinary behavior in communicating events between documents, such as the delayed delivery of a subscribed event, builds the ground for these situations. Exceptional situations can be anticipated in schema design and should be dealt with by exception rules, providing a meaningful design distinction. No additional concepts are necessary to model exception rules. They react to specific logical events, which determine exceptional situations, and trigger necessary corrections.

Example 8. Figure 3 shows event class `pubDeliveredLate`, which collects exceptional situations, i.e. `published` events that are delivered late. Exception rule `redetermineActResRule` (cf. snippet 2) defined on `pubDeliveredLate` reacts to the late delivery by redetermining the researcher’s active researcher status by invoking operation `publishActResStatus`.

4 Managing Active XML Schemas

To conclude this paper, we briefly discuss the management of active XML Schema documents. A detailed treatment is beyond the scope and space limitations of this paper.

4.1 Reactive Service

Since XML documents are static by nature, they need to be complemented by a *reactive service* that handles the presented behavioral extensions (for passive behavior as well as active behavior). Various components of a reactive service, such as an event detector and a rule manager, can be built upon experience from active database systems. The management of simple active XML rules based on mutation events has been already discussed in [5]. Due to the higher-level approach taken with active XML Schemas, new issues arise, which we sketch in the following.

First, the occurrence of an event does not correspond to the addition of an event to an event class, but to the event’s occurrence time matching wall clock time (system time), since event classes may contain scheduled events. Second, the delivery time of a remote event needs not be past the event’s occurrence time, because a remote event may be delivered to subscribing documents before its occurrence time. This is similar to employers delivering salary payment records ahead of the payment day to employees’ bank accounts, which also list such transactions on account statements with a future effective date (occurrence time).

Third, events need not be added to a document’s event classes when they occur. When a document is checked out from a reactive service to be edited in a legacy application, an edit script as a collection of mutation events (cf. [5]) reflecting the document changes is added to the document only when it is checked in again. In a similar way, subscribed remote events may be collected by a behavior service in an

“in-box” before these events are added to a document. All these events, which are added delayed, should be added to a document’s event classes before any rule is triggered.

Fourth, as documents may not be accessed for some time, event occurrences need not to trigger rules immediately (*eager mode*), but may trigger rules only before a document is accessed next time (*lazy mode*). In lazy mode or when events are added delayed, rules can inquire by querying event classes what other events are “pending”. This approach may be compared to skimming all e-mails in a mail box before answering in order to avoid a reply that would have been different if an already delivered later e-mail had been read. This enables designers to realize a “net effect” policy as known from active databases [18].

Fifth, rules may also need to react according to events reflecting active behavior, i.e. rule executions, complementary to events reflecting passive behavior, i.e. operation executions. To inquiry the status of rule executions, for each event and each rule defined on the event’s class a *rule trace* records whether the rule has not yet been processed, is being processed, or has been processed successfully. For easy checking whether all rules that are defined on an event class have been considered or processed for a given member event, each event provides information reflecting its *event status*. The status of an event is “occurred” after its addition and it is “processed” after all rules defined on the event class the event is member of, have been considered. An event that occurs in the future, but is already added to an event class, shows status “scheduled”.

4.2 Communication Transparency

Import and export of event classes between documents is specified declaratively, disregarding the delivery approach chosen. To provide for communication transparency, the delivery of subscribed events is delegated to a *publication handler* associated with the exporting document and a *subscription handler* associated with the importing document. The delivery approach may be specified separately for each document type or document and states whether a pull or pushed-based approach is taken, what the polling period is in the case of a pull-based approach, how delivered events are buffered, or whether events are pre-filtered by the publication handler if not all events of an exported event class are needed by the subscribing document (techniques outlined in [6] for pushing reactive services to the XML repositories may be employed for pre-filtering). Similar to physical data independence, which hides changes to the internal storage structure of a database from database applications, communication transparency hides changes from the delivery semantics from event classes and rules.

4.3 Event Source Transparency

Events that occur within legacy applications may as well as active XML documents give rise to active behavior. Therefore, and because an active XML Schema should be shielded from changes to its event sources, we provide for event source transparency by which changes to event sources are hidden from event classes and rules.

For example, an active XML document may contain data drawn from a data-base. Database changes need to be propagated to the document. In such a situation, an active XML Schemas adapter can be defined for the database, which exports appropriate event classes reflecting database updates.

Acknowledgment. We thank Thomas Thalhammer for carefully reading the paper and giving helpful comments. This research was carried out partly at the ACRC, University of Australia, during August-September 2001.

References

1. ebXML. <http://ebxml.org>, 2001.
2. BizTalk Framework. <http://biztalk.org>, 2001.
3. S. Abiteboul, et al.. Active Views for Electronic Commerce. In *Proc. VLDB*, 1999.
4. P. V. Biron and A. Malhotra. XML Schema Part 2: Datatypes. <http://w3.org/TR/xmlschema-2>, 2001.
5. A. Bonifati, S. Ceri, and S. Paraboschi. Active Rules for XML: A New Paradigma for E-Services. *VLDB Journal*, 2001.
6. A. Bonifati, S. Ceri, and S. Paraboschi. Pushing Reactive Services to XML Repositories using Active Rules. In *Proc. WWW10*, 2001.
7. T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. Extensible Markup Language (XML) 1.0. <http://w3.org/TR/REC-xml>, 2000.
8. M. Carey, et al.. XPERANTO: Publishing Object-Relational Data as XML. In *Proc. WebDB*, 2000.
9. Sharma Chakravarthy. SENTINEL: An Object-Oriented DBMS With Event-Based Rules. In *Proc. ACM SIGMOD*, 1997.
10. J. Clark. XSL Transformations (XSLT) Version 1.0. <http://w3.org/TR/xslt>, 1999.
11. J. Clark and M. Murata. RELAX NG. <http://oasis-open.org/committees/relax-ng>, 2001.
12. M. Fernández, A. Morishima, and D. Suciu. Publishing Relational Data in XML: the SilkRoute Approach. *IEEE Data Engineering Bulletin Board*, 24(2), 2001.
13. H. Ishikawa, et al.. Project Xanadu: XML- and Active-Database-Unified Approach to Distributed E-Commerce. In *Proc. DEXA Workshop*, 2001.
14. R. Jelliffe. The Schematron. <http://www.ascc.net/xml/resource/schematron>, 2001.
15. P. Lang, W. Obermair, and M. Schre. Modeling Business Rules with Situation/Activation Diagrams. In *Proc. ICDE*, 1997.
16. S. Lechner, M. Schrefl, and G. Preuner. Translating XQuery into XSLT. In *Proc. DASWIS*, 2001.
17. K. J. Lieberherr. Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns. PWS Publishing Company, 1996.
18. N. Paton and O. Diaz. Active database systems. *ACM Computing Surveys*, 31(1), 1999.
19. T. Pixley. Document Object Model (DOM) Level 2 Events Specification. <http://w3.org/TR/DOM-Level-2-Events>, 2000.
20. M. Reinhold. The Java Architecture for XML Binding (JAXB). <http://java.sun.com/xml/jaxb>, 2001.
21. J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison Wesley, 1999.
22. M. Schrefl, et al.. Self-Maintaining Web Pages - An Overview. In *Proc. ADC*, 2001.
23. G. Sindoni. Incremental Maintenance of Hypertext Views. In *Proc. WebDB*, 1998. H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn. XML Schema Part 1: Structures. <http://w3.org/TR/xmlschema-1>, 2001.

27. M. Toyama and T. Nagafuji. Dynamic and Structured Presentation of Database Contents on the Web. In *Proc. EDBT*, 1998.
28. C. M. Vittori, C. F. Dorneles, and C. A. Heuser. Creating XML Documents from Relational Data Sources. In *Proc. EC-Web*, 2001.
29. X. Zhang, et al.. Clock: Synchronizing Internal Relational Storage with External XML Documents. In *Proc. RIDE Workshop*, 2001.

Behavior Abstraction in Semantic B2B Integration

Christoph Bussler

Oracle Corporation, Redwood Shores, CA 94065, U. S. A.
Chris.Bussler@Oracle.com

Abstract. Trading partners exchanging business messages follow each a pre-defined protocol called a public process (“public”, because its definition is publicly available). Internal processing of received or to be sent business messages is done through private processes that are invisible to trading partners. Public processes are quite complex because they rely on inherently insecure and unreliable networks. Furthermore, they need to be linked with private processes without revealing their internal logic. This paper introduces an advanced model for defining public processes based on a set of elaborate requirements.

1 Introduction

The goal of semantic B2B integration is to connect companies with their trading partners electronically through organized message exchanges containing business data in order to conduct business. Ideally, all message interactions take place over networks like the Internet [16] or VANs [33] and do not require any human intervention at all. An example is the automatic exchange of a purchase order (PO) and a purchase order acknowledgment (POA) message between two companies. Once a particular PO is sent from a buying company (buyer) to a selling company (seller) a POA has to be sent back from the seller to the buyer acknowledging (or rejecting) the purchase order (see Fig. 1).

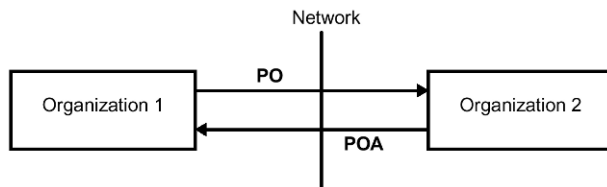


Fig. 1. PO-POA round-trip document exchange between two enterprises

A naïve approach is to implement the processes within each trading partner as one “cooperative” workflow [7] (see Fig. 2). If one workflow sends a message, the other one receives it. The workflows contain the logic for the message exchange (the public process), the business logic as well as how documents are inserted or retrieved from back end systems (the private process).

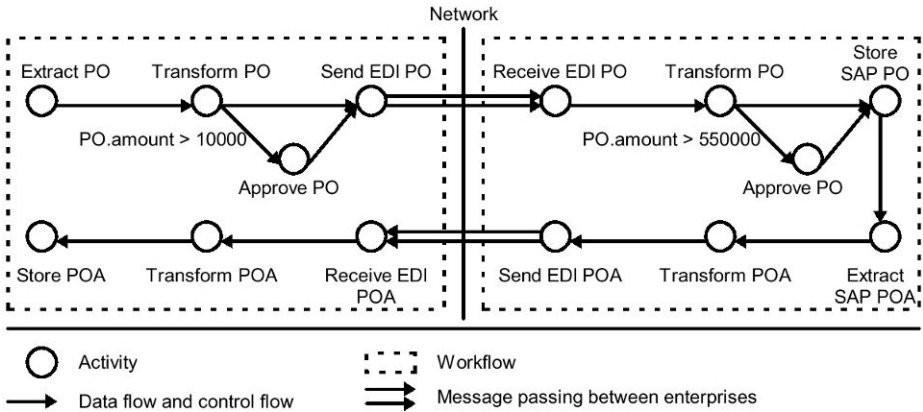


Fig. 2. “Cooperative” workflows

This approach has several major deficiencies related to public and private processes (and therefore it is attributed “naïve”):

- The message send and message receive sequencing for a given message exchange is encoded in the workflow itself. If several workflow types use the same sequencing they have to implement the sequencing repeatedly.
- The required transformations between the document formats of the back end applications and those of the B2B protocol is performed within each workflow type. The workflow modeler has to place the transformation correctly for all possible execution paths.
- It is not possible to make the message exchange behavior public without revealing the whole workflow.

An advanced approach that is advocated in this paper is to separate the message exchange behavior (public process) from the business logic (private process). This results in several processes that each has its focus. Public processes define the message exchange behavior over a network whereas private processes (workflows) define the trading partner internal business logic processing (see Fig. 3) [7].

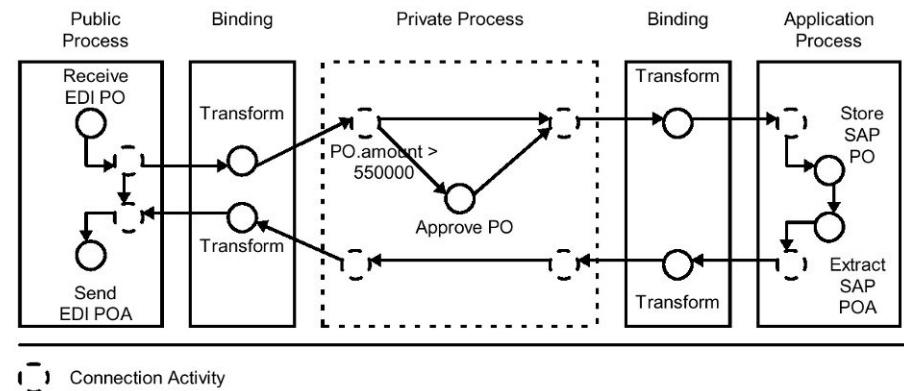


Fig. 3. Public and private processes

In Fig. 3 the public and the private process are bound through bindings that make sure that the same public process can be reused with different private processes through different bindings. Bindings use connection activities to bind public and private processes. Also, the same private processes can be bound to different public processes (if a trading partner communicates with several other trading partners that follow different public processes) by means of different bindings [7]. Fig. 3 implements the right side of Fig. 2. The transformation within the binding allows to abstract from the particular format required or sent by the B2B protocols or back end applications. This way it is possible that private processes work on a normalized format.

This approach overcomes the deficiencies enumerated above since the private processes (or workflows) do not have to model the message exchange sequences any more, the private processes do not have to do the transformation any more and the public as well as private processes are modeled once and can be reused.

The contribution of this paper is a detailed analysis of the requirements for public processes and the construction of an advanced model for public process definition and execution. Section 2 is an elaborate analysis of requirements for public processes. This analysis goes into many functional and technical details in order to cover all aspects of public processes. Section 3 presents an advanced model that supports the definition and execution of public processes. Section 4 discusses related work.

2 Requirements for a Public Process Model

Public processes are a sophisticated concept that requires careful requirements analysis and model design. This section is devoted to the analysis of public processes in detail that support semantic B2B integration.

2.1 Message Types

There are several message types that can be distinguished. From a public process perspective they are all messages, however, from a business perspective they have different meaning and are of different importance.

- **Business message.** Business messages are messages carrying business content like an EDI 850 transaction set (purchase order in the EDI standard [11]). The content of business messages is usually inserted or retrieved from a back end system like an ERP.
- **Acknowledgment message.** Acknowledgment messages confirm the receipt of business messages. For example, the EDI 997 transaction set (functional acknowledgment) is sent to indicate that a business message was successfully received.
- **Administration message.** An administration message carries information that is not related to business content itself. For example, a trading partner message like the EDI 838 transaction set (trading partner profile) is sent to update trading partner specific information like phone numbers or addresses.

- **Error message.** An error messages like the EDI 842 transaction set (non-conformance report) is used to indicate errors in business messages. This allows the sending trading partner to investigate why inconsistent content was sent.

In the following messages are treated alike except when explicitly noted. This guarantees that the implementation of public processes is uniform across the message types and only the semantics differs for the different message types.

2.2 Message Patterns

The example in Fig. 2 sends one PO and expects back one POA. This is a particular message pattern where one business document is sent and one business document is returned. What is not shown but always required (by standards like ebXML [8] or RosettaNet [29]) is that each business message is acknowledged by an acknowledgment message. The PO-POA exchange is called a (business) message pattern whereby the underlying acknowledgment message exchange is implicitly assumed and not explicitly mentioned.

- **One-way.** One business message is sent and no business message is returned.
- **1 request - 1 reply.** One business message is sent and one business message is returned. This is the example in Fig. 2, the PO-POA round-trip.
- **1 request - n replies.** In this case a business message is sent and several business messages are expected as return. An example is a PO being sent whereby several POAs are returned confirming individual line items of the PO.
- **n requests - 1 reply.** Several business messages are send and one is returned. An example is a PO sent in pieces that is confirmed with one POA.
- **n requests - m replies.** This case is the combination of the two previous cases.
- **1 subscribe - n publish.** A business message is sent that is taken by the receiver as a subscription for information. The receiver can publish back to the sender whenever relevant data becomes available. In this case none, one or several replies can be sent back.

2.3 Public Processes

Public processes implement one or more message patterns. For example, in Fig. 2 one message pattern was defined, the PO-POA request-reply pattern. However, sometimes more patterns have to be composed in order to achieve a working message exchange. For example, in the subscribe-publish pattern another pattern is required to unsubscribe. That second one might be a one-way pattern. Both patterns together make sure that a subscription can be canceled. Fig. 4 shows this example. In the upper portion the subscribe/publish behavior is shown. A subscription is sent and publications are received. There might be any number of publications. This is possible by the loop on the sending side (right side) as well as receiving side (left side).

The representation in Fig. 4 shows that a public process has two halves, one for each trading partner involved. The left side in Fig. 4 defines the behavior of the

subscribing trading partner whereby the right side defines the behavior of the publishing trading partner. Both halves are necessary to define the complete public process between the two trading partners.

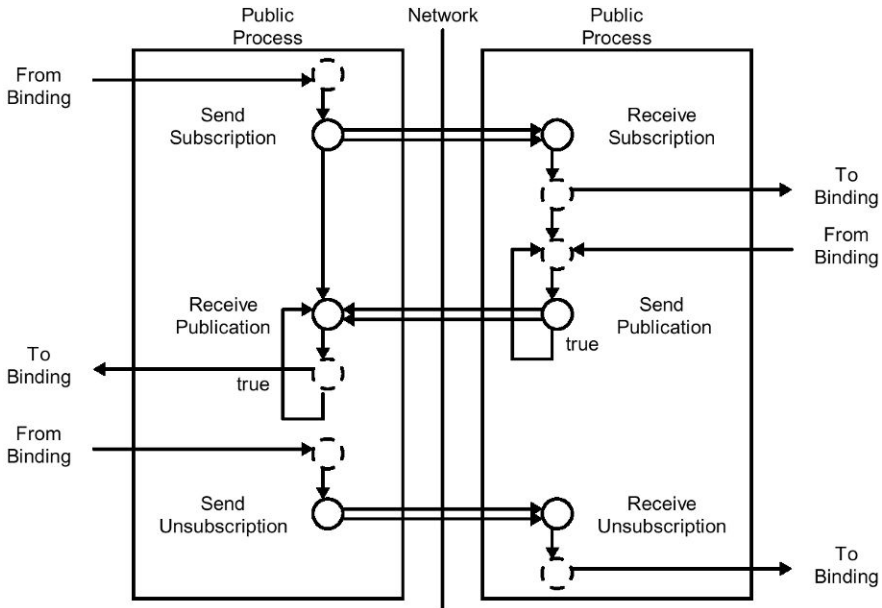


Fig. 4. Public protocol with two message patterns

Public processes are not limited to two trading partners. It is possible to have three or more trading partners involved. This is also called multi-party public processes (or multi-party collaborations in ebXML [9] [10]). Multi-party public processes can be represented as several two-party public processes. In the following, multi-party public processes are not discussed.

The two halves involved in a public process must be consistent wrt. to the message exchange. This means that for every message sent by one trading partner there must be an activity in the other trading partner's half to receive the message. Only then it is possible to exchange messages in lockstep. However, for every receive there does not have to be a sent. For example, in a request for quotation (RFQ) it is not mandatory to respond to a request for quote. This means that the trading partner that sends requests for quotes might or might not get quotes back. Quotes can come at any time until he makes a decision selecting one from the received quotes. Afterwards, no quotes are accepted any more.

2.4 Control Flow in Public Processes

Public processes require several types of activities in order to define message exchange behavior. Sending and receiving activities are used for message transmission

with trading partners. Entry point activities are used to send a message to an ongoing public process in parallel to the control flow in the public process. Connection activities are required to allow a binding from a public process to a private process (see Fig. 3 and [7]). Time-out activities are activities that wait for a given time and finish automatically after that. This allows to implement time-out behavior.

Activities have to be connected by control flow constructs that define their order of execution. Sequential and conditional control flow allows the basic ordering. However, in order to implement retry behavior a looping construct is required that can execute a set of activities again after a time-out activity. A selective parallelism is necessary in order to implement the fact that two activities can be executed without being sure which one will be finished. For example, when a business message is sent either a response message is received or a time-out happens to re-send. This means that two activities are started in parallel but only one is finished.

Another example for selective parallelism is a lost acknowledgment message. In this case a retry happens even though the trading partner that sent the acknowledgment message assumes that the acknowledgment was transmitted successfully. In this situation the original business document is sent again and must be received in parallel to the ongoing processing in order to dismiss the duplicate.

Private and public processes are sometimes ongoing in parallel. For example, in Fig. 4, once a publication is received, it is given to the binding through a connection activity and in parallel the loop is going back to the “receive publication” activity. This is a parallel split that causes independent branches to go on without ever joining again. On the other side in Fig. 4 a parallel join takes place at the connection activity before the “send publication” activity. At this connection activity the public process internal flow and the flow coming from the binding are joined. These join from two different directions without ever branched before. This is a parallel join.

Business messages have in many cases a relationship to each other. For example, a PO request and a PO cancellation are related if they refer to the same PO. However, a PO request transmission can happen in a separate public process from that of the PO cancellation. If there is a public process for each of the two documents then both public processes are executed separately. This means that they are bound independently to private processes. If this is the case it is not known that two related business messages are processed. This is called an out-of-bound situation. This is not bad situation in itself. What happens is that both messages are sent to a back end system and the back end system first creates the purchase order and then cancels it. This happens even if the PO cancellation is sent right and immediately after the PO creation. The only requirement is that the semantic integration server preserves the order such that if the PO request is received first it is first inserted into the back end application. Otherwise it would cause an error situation in the back end system because if the cancellation would be first, the back end application would reject it due to “no PO found” error. Then the PO request arrives and it would be accepted as a valid document.

If both, the PO request as well as the PO cancellation would be defined within one public process, the situation would be different. In this case it would be possible to stop the PO request before it is inserted in the back end application system. This is

possible since it would be possible to first stop the private process and then figure out if the PO request was already sent to the back end. If not, it could be neutralized before even being inserted into the back end system by the arriving PO cancellation. An entry point for the PO cancellation allows this.

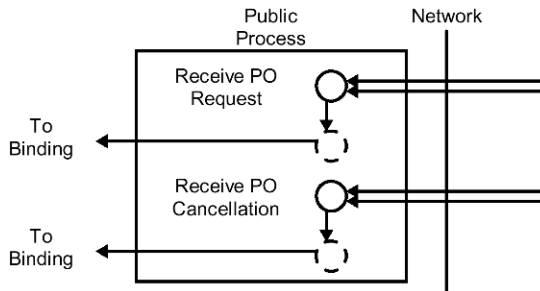


Fig. 5. Out-of-bound processing in public processes

Fig. 5 shows the situation. The PO cancellation can be received at any time and the binding that receives the PO cancellation can stop the private process to determine if the PO request is already processed by a back end application.

2.5 Order Preservation, Correlation, and Duplicate Detection

- **Order Preservation.** Since it is in general possible that out-of-bound business messages exist that are processed by different public processes, a semantic B2B integration server must make sure that the order of the incoming messages as well as the outgoing messages is preserved so that the sequence of the messages does not change. This makes sure that the environment does not introduce errors due to message sequence re-ordering (see example above with the PO request and the PO cancellation).

Some B2B protocols have a field in the message header that indicates the sequence number of a message. This sequence number can be used to check if the messages come in the correct order (i. e. according to the sequence the sequence number expresses). In this situation the integration server can re-order messages so that they are ordered correctly according to the sequence number.

- **Message Correlation.** When several business messages of the same type (like PO) are sent that all require a business message response (POA) then the question arises which POA is correlated to which PO so that the appropriate instance of the public process can be continued. There are two fundamental possibilities that have to be supported both since B2B protocols require them. One is that the messages are correlated by some field that contains business data. For example, a PO contains a unique PO identifier and the POA refers to that. This means that the correlation between two documents is given if the PO identifiers match. Based on this

match the appropriate instance of the public process can be selected and continued.

The other possibility is that the B2B protocol carries an unique identifier in the protocol specific header information (that is outside the business content). It is assumed that the same unique identifier is used in the header when the response message is sent. In this case the correlation is based on unique identifiers in the protocol headers, not in the document contents. Since the message is sent from a send activity the unique identifier also relates to exactly one public process. RosettaNet [29] follows this approach.

- **Message Duplicate Detection.** If a message needs to be resent due to a time-out, then either the original business message was lost during transmission or the acknowledgment back to the sender. If the latter is the case and the business document is re-sent the receiver receives the same message twice. However, in this case the processing of this message should not happen twice. Instead, the duplicate message should be discarded. It is therefore mandatory that duplicate messages are caught and discarded so that the public process and the subsequent private process do not process them.

3 Advanced Model for Public Processes

The modeling constructs required for public process modeling are introduced in the following.

3.1 Messages

Messages are modeled as data types with a well-defined internal structure. This allows structured access to the message content so that it is possible to determine the type of a message, the instance identifier of a messages as well as individual fields like “amount” in an instance of a “PO” (“PO.amount”). (Security like encryption as well as transport considerations are not discussed here).

3.2 Activities

The following types of activities are necessary to define public processes.

- **Send.** A send activity can send messages to trading partners. Send activities can indicate which type of message they can send to trading partners.
- **Receive.** A receive activity can receive messages from trading partners. A receive activity can indicate which type of message it can receive by defining a message type. If used in a loop construct one receive activity can receive several message instances sequentially.
- **Entry point.** An entry point is a receive activity that can receive message asynchronously to the ongoing control flow in a public process. For example, in an

out-of-bound situation an entry point can receive a message while message processing is going on concurrently in the public process. A special join construct (“parallel join”, see below) allows the independently started control flow by an entry point to join ongoing control flows in public process.

- **Connection.** Connection activities are activities that can be bound by bindings and represent the interface of a public process for the binding. Message instances can be sent to the public process through connection activities or sent to the binding through connection activities. The control flow either comes from the binding to the public process or comes from connection activities in the public process to bindings. Accordingly, connection activities are incoming or outgoing.
- **Consumption.** A consumption activity is a activity that is a valid end point in the control flow within a public process. For example, if the public process detects a duplicate business content then this message might not be further processed. In this case the part of the public process that detected the duplicate business content can finish with a consumption activity with no outgoing control flow.
- **Time-out.** A time-out activity will start and wait for the indicated time until it finishes. After that the control flow after this activity continues.

In a graphical modeling tool all these activities would be represented through different icons.

3.3 Control Flow

The following control flow constructs are necessary to model public processes. These are

- **Sequence.** A sequence between two activities means that once the first activity is executed the second one will be started.
- **Conditional branching.** A conditional branch consists of a Boolean expression that might refer to message contents. The outcome of the Boolean expression is a truth value of either true or false. For each of the outcomes a next activity is modeled that is executed based on the outcome.
- **Loop.** A loop allows to repeatedly execute one or more activities in the loop body. A loop has a condition in form of a Boolean expression that indicates if the loop body should be executed. If the expression evaluates to true, the body is executed again.
- **Selective parallelism.** The selective parallelism starts several activities in parallel. However, only one of the activities can succeed. As soon as one succeeds all the other activities are finished immediately. This allows to model a situation where it cannot be determined which of a set of activities will succeed.
- **Parallel.** A parallel construct starts several activities in parallel and unconditionally. This means that the activities are expected to execute independently from each other. Once all the parallel activities are finished, they are joined together again by a join construct (see below).

- **Parallel split.** A parallel split starts several activities in parallel without them joining back later in the control flow within the public process. This allows to send a message to a binding while the processing in a public process is ongoing.
- **Join.** A join constructs joins parallel control flows that have been started by a parallel construct (see above).
- **Parallel join.** The parallel join construct can join two control flows that have not a common split point before, i.e. before the parallel join, the control flows are independent. This is different from the normal join in that a normal join assumes that a parallel construct was executed.

3.4 Data Flow

The data flow within public processes is based on message instances. Each message instance that is processed in a public process is available to all activities in the public process. Several message instances can be in a public process concurrently. This means that the documents must be individually accessible. This is achieved by introducing the concept of typed and named local message variables that is similar to local variables in programming languages. A receiving activity or a receiving connection activity can assign the message instance to a local message variable. This is done by value so that the message instance content is available as long as the public process exists.

3.5 Execution Model

Public processes follow a type/instance approach. Modeled public processes are types that can be instantiated any number of times. This allows to execute any number of message exchanges in parallel with trading partners. Each instance of a public process has a unique identifier that allows to correlate back response messages to the appropriate public processes. This allows to complete message exchanges within public processes completely.

Incoming messages (from bindings or from trading partners) can either start or continue public processes. Response messages, acknowledgment messages as well as error message continue public processes since they are in response to a business message sent earlier. Business message might continue a public process (like the POA example) or start a new public process instance. For example, if a seller receives a PO then this PO causes a new public process that starts processing the PO.

When a message continues a public process then it is possible to find out based on the correlation (see Section 2.6) of the messages to which public process they belong since a public process knows the identifier of the messages sent out.

In summary, the execution model is similar to those of type/instance based workflow management systems [18].

4 Related Work

Two examples of standards addressing public processes are RosettaNet [29] and ebXML [8]. RosettaNet pre-defines specific public processes called Partner Interface Processes (PIPs) whereby ebXML allows to define arbitrary public processes through Collaborations. This means that in case of RosettaNet two enterprises can inter-operate by virtue of being RosettaNet compliant. For example, the round-trip example of a PO-POA exchanged is implemented in a PIP called 3A4. In the case of ebXML two enterprises have to agree on a definition of their public processes first. This is facilitated by Collaboration Partner Agreements (CPAs) [10]. Standards like EDI [11] are neither defining public processes nor providing a mechanism to define public processes. In this case enterprises need to borrow the mechanism from e. g. ebXML to define public processes. None of the standards addresses the interplay between public and private processes. Neither RosettaNet nor ebXML pre-define specific activity types of public processes like send/receive, connection, etc.

Crossflow [12] is concerned about a methodology to establish and to maintain contracts between enterprises. Furthermore, a system is proposed to enforce the contracts. However, the system is not built to deal with arbitrary public processes and no standard definition language and semantics is provided for the enforcement of contracts between two enterprises. All enterprises involved are required to use the same software for contract enforcement.

WSFL [37] is concerned about a flow language describing web services composition. It is not concerned about the definition of public processes. The same applies to XLANG [38] and BPML [4].

Products of vendors like [3] [17] [20] [28] [32] [34] [35] [36] support inter-enterprise message exchange following specific selected standards. However, it is impossible to derive from the data made available by the vendors how the internal implementation would support the advanced modeling approach for public processes proposed in this paper.

In [1] an approach is presented whereby the private process inherits from the public process through workflow inheritance [5]. The inheritance makes sure that all activities of the public process are available in the private process. The private process can extend the public process with internal workflow activities. The problems with this approach are the same to those discussed in Section 1 for the cooperative workflow approach. This is the case because the private process contains the message exchange logic from the public process. In [1] this is achieved by inheritance whereby in Section 1 this was designed into the workflow without inheritance. No matter how the public process comes into the private process, the fact is that the private process contains the public process. Therefore all problems listed in Section 1 apply. In addition, out-of-bound messages, duplicates and other requirements are not addressed in [1].

References

1. Aalst, W. M. P. van der; Weske, M.: The P2P Approach to Interorganizational Workflows. In: *Proceedings of the 13th International Conference on Advanced Information Systems Engineering* (CAiSE'01), Vol. 2068 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2001.
2. BEA Weblogic. www.bea.com/products/weblogic/server/index.shtml
3. BEA Weblogic Integration. www.bea.com/products/weblogic/integration/index.shtml
4. BPML. www.bpmi.org
5. Bussler, C.: Workflow Class Inheritance and Dynamic Workflow Class Binding. In: *Proceedings of the Workshop Software Architectures for Business Process Management* at the 11th Conference on Advanced Information Systems Engineering CAiSE*99, Heidelberg, Germany, June 1999
6. Bussler, C.: B2B Protocol Standards and their Role in Semantic B2B Integration Engines. In: *Bulletin of the Technical Committee on Data Engineering*. March 2001, Vol. 24, No.1. IEEE Computer Society
7. Bussler, C.: The Role of B2B Protocols in Inter-enterprise Process Execution. In: *Proceedings of the Workshop on Technologies for E-Services* (TES 2001), Rome, Italy, September 2001
8. ebXML. www.ebxml.org
9. ebXML Business Process Specification Schema. Version 1.01. May 2001. www.ebxml.org
10. ebXML Collaboration-Protocol Profile and Agreement Specification. Version 1.0. May 2001. www.ebxml.org
11. EDI. www.x12.org
12. Hoffner, Y.; Field, S.; Grefen, P.; Ludwig, H.: *Contract-Driven Creation and Operation of Virtual Enterprises*. Research Report, RZ 3328, #93374, Computer Science, IBM Research, February 2001
13. IBM DB2. www.ibm.com/db2
14. IBM MQ Series. www-4.ibm.com/software/ts/mqseries
15. IBM Websphere. www.ibm.com/websphere
16. Internet. webopedia.internet.com/TERM/I/Internet.html
17. IONA Netfish XDI. www.iona.com/products/
18. Jablonski, S.; Bussler, C.: *Workflow Management. Concepts, Architecture and Implementation*. International Thomson Publisher (1995)
19. Java. java.sun.com/
20. Microsoft Biztalk Server. www.microsoft.com/biztalkserver
21. Microsoft MQ. www.microsoft.com/msmq/
22. Microsoft SQL Server. www.microsoft.com/sql/default.htm
23. OAGIS. www.openapplications.org/
24. Oracle Advanced Queuing. technet.oracle.com/docs/products/oracle8i/doc_library/817_doc/appdev.817/a76938/toc.htm
25. Oracle Application Server. www.oracle.com/ip/deploy/ias/index.html
26. Oracle eBusinessSuite. www.oracle.com/applications/index.html?content.html
27. Oracle Database. www.oracle.com/ip/deploy/database/index.html
28. Peregrine B2B Integration Platform. www.peregrine.com
29. RosettaNet. www.rosettanet.org
30. SAP. www.sap.com

31. The XML Cover Pages. www.oasis-open.org/cover
32. TIBCO Active Exchange. www.tibco.com/products/activeexchange
33. VAN. www.edi-info-center.com/html/vans.html
34. Versata Integration Server. www.versata.com
35. Vitria Business Ware. www.vitria.com/products/businessware/overview.html
36. WebMethods B2Bi. www.webmethods.com/content/1,1107,B2BiSolutions,FF.html
37. WSFL. *Web Services Flow Language*. Version 1.0. IBM Software Group, May 2001. www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
38. XLANG. www.gotdotnet.com/team/xml_wsspecs/

OIL Ontologies for Collaborative Task Performance in Coalitions of Self-Interested Actors

Vadim Ermolayev, Natalya Keberle, and Vyachyslav Tolok

Dept. of Mathematical Modelling and IT, Zaporozhye State Univ.,
66 Zhukovskogo st., 69063, Zaporozhye, Ukraine
{eva, kenga, tolok}@zsu.zp.ua

Abstract. Presented are the Task Model and the ontologies for arranging co-operative work in an open organization of intelligent agents-executives. These agents dynamically form the coalitions for collaborative task performance. Coalition formation is guided by contracting negotiation in frame of the Arrangement Phase. The role of the Task and Negotiation Ontologies is to provide the shared conceptualisation of the terms, the structures and the procedures used by agents in the processes of activity analysis, decomposition, performance and delegation. The ontologies are formalized in OIL and are translated to DAML (RDF), RDFS, SHIQ notations, thus providing the concepts in the forms of emerging service mark-up standards.

1 Introduction

The development of the means for semantic interoperation among intelligent service providing agents is one of the mainstreams of the Semantic Web [1] implementation activities. As it was mentioned in [2], “The Web once solely a repository for text and images is evolving into a provider of services. ... we are seeing increased automation of Web Service interoperation primarily in B2B and E-Commerce applications. ... Fundamental to reliable, large-scale interoperation of Web services by computer programs or agents is the need to make Web Services directly understandable – to create a Semantic Web of services whose properties, capabilities, interfaces and effects are encoded in an unambiguous, machine-interpretable form.” Clearly, these services may be formally seen as distributed flows of activities among intelligent, self-motivated and benevolent actors. These actors are both humans and software agents.

Distributed processes, dynamic changes in process flow, self-interested behaviours and collaborative performance are obviously the intrinsic features of the E-Commerce world. Contemporary requirements to software implementations in the domain are, thus, the kind of the intelligent equilibrium on these, sometimes contradictory, features. Economically rational [3] artificial actors performing intelligent activities and tasks are often implemented as software agents. Distributed interoperable implementations are based on sharing knowledge among the open organization [4] of the actors on how to perform an activity or a cluster of activities in rational collaboration with each other. The ontologies providing conceptualisations for knowledge sharing among the agents and for rational activity arrangements are the

focus of this paper. These ontologies are used in the process of co-ordination of collaborative performance in dynamic coalitions [5] of self-interested agents-executives.

An important aspect is that co-ordination patterns in the domain should be as real-life as possible taking into account the balance of the actors' features of rationality, self-interest and benevolence (in the sense of being rationally ready to collaborative performance). In frame of the presented research in progress this goal is approached by the ability of the actors to dynamically form the coalitions for optimal task execution. Coalition formation is co-ordinated by negotiation.

An actor may either be not able or not really interested to perform the whole task itself due to various reasons: lack of capability, overload, self-interest. One of the democratic and the rational ways for an actor to find collaborators for performing parts of the task is to negotiate with its fellow actors and to choose the optimal bid presented by an optimal trade-off. Negotiation pattern used in the presented approach is based on the introduction of the so-called Arrangement Phase for negotiation on activity placement. Negotiated are trade-off functions providing the dependencies of the proposed incentive over time.

The remainder of the paper is organised as follows: Section 2 is devoted to related work; Section 3 introduces the motivations of the research by presenting an example of a perspective B2B investment consulting electronic marketplace; Section 4 briefly outlines the Task Model for a community of self-interested actors; Section 5 sketches out the framework for negotiation on activity placement within the Arrangement Phase; Section 6 presents the Task Ontology and the Negotiation Ontology; Section 7 presents the conclusions and the plans for future work.

2 Related Work

To weave shared semantics into the Web with its intrinsic aspiration for distributedness, autonomy democracy and privacy is quite a hard issue. As Makiavelli mentioned in 1513, there is no cause more difficult in conception, more dubious in success, more dangerous in implementation than introducing a new order. Substantial effort is applied to resolve the problem by various research communities as well as by standardization bodies trying to unify the means for concept sharing and semantic interoperability, co-ordination patterns, process description and service provision facilities. In distributed process management and service provision domains the activities in these 3 streams may be presented as follows.

The major standardization effort in process, workflow modeling and management belongs to WfMC¹. The major accomplishments of WfMC in the field are: the Process Model – workflow and activity representation, XPD L – XML binding of PDL. Semantic interoperability solution is generally seen by WfMC as Workflow-XML binding.

As it was mentioned in [6], "...cooperation and coordination of the planning, monitoring and workflow of the organizations can be assisted by having a clear shared model of what comprises plans, processes and activities...". Known are the efforts aiming to define the basic shared concepts: SPAR [6] ontology, the Enterprise

¹ Workfolow Management Coalition. <http://www.wfmc.org/>. Last accessed on Nov. 4, 2001

Ontology [7], Process Specification Language [8] (ordered hierarchies of activities), ToVE [9] (shared terminology for a virtual enterprise), aggregation of activities at multiply levels [10], O-Plan research [11] (manipulating plans of task execution) and others.

A comprehensive survey of negotiation approaches to distributed service provision may be found, for example, in [12]. In E-Commerce, for instance, the models for coalition formation based on pre- and post- negotiation patterns are proposed in [13] having COALA² as the general-purpose testbed for studying co-operative behaviours in agent coalitions. The approach of the presented research is close to that of service oriented negotiation [14], which involves determining a contract under certain terms and conditions. The mechanisms of service provision via negotiation discussed in this paper are more close to that of ADEPT [15], though, more authoritarian algorithms are investigated as well [16]. Alternatively, the capability based approach to service matchmaking (Eg., [17]) is proposed to determine the proper candidate to become the service provider.

Substantial results are appearing in the development of the languages for service mark-up. DAML+OIL [2] initiative promises probably the best perspectives as both languages are built on top of RDF(S) – W3C³ metadata standard.

3 Motivating Example

Authors' earlier publications addressed several modelling cases of project planning [18], personnel (PhD) recruiting [19], production management [20] processes. These case studies used simplified modelling methodology. The enhancement of the modelling framework was primarily motivated by the desire to model more sophisticated collaborative interrelationships and behaviours, than 100 per cent altruistic commitment to gain the organizational goal [20]. The following example of a possible B2B Consulting E-Market application may be convincing in the necessity for the actor agents to be more economically rational as well as to simultaneously possess substantial willingness to collaborative performance.

Suppose ABC is a successful consulting company working in the field of Capital Construction Investment (Fig. 1). Assume ABC organization comprises the following staff of actors: Project Managers, Construction companies' representatives, Construction materials supply companies' representatives, Transportation companies' representatives, Community officials' representatives. Each of the actors presents the capabilities and the interests of the "wrapped" organizations. From the other hand, it is ABC fellow member and should be concerned about company's success and revenues. The environment, ABC works within, is inhabited by the perspective investors. The investors seek for effective investments in the field.

To be successful on the market ABC needs to provide attractive investment plans with minimal risks. These investment plans should balance on the mutual interests of both the investors and the represented companies from one hand as well as upon the constraints on the resources and on the executives involved. The investors and the

² <http://www.cs.cmu.edu/~softagents/coala.html>. Last accessed on Apr. 10, 2001.

³ World Wide Web Consortium. <http://www.w3c.org>. Last accessed on Nov. 4, 2001.

wrapped executives may also have overlapping, conflicting and/or coherent interests, commitments and intentions (Fig. 1).

To provide credible investment proposals in response to the investors' queries ABC should be capable to model (simulate and evaluate) the processes of corresponding projects' implementation, reason about the possible behaviour of the executive organizations participating in the project and reason about the risks as well as about the possible overall project success or failure. It should also provide reasonable recommendations on corrective influences for the critical project steps.

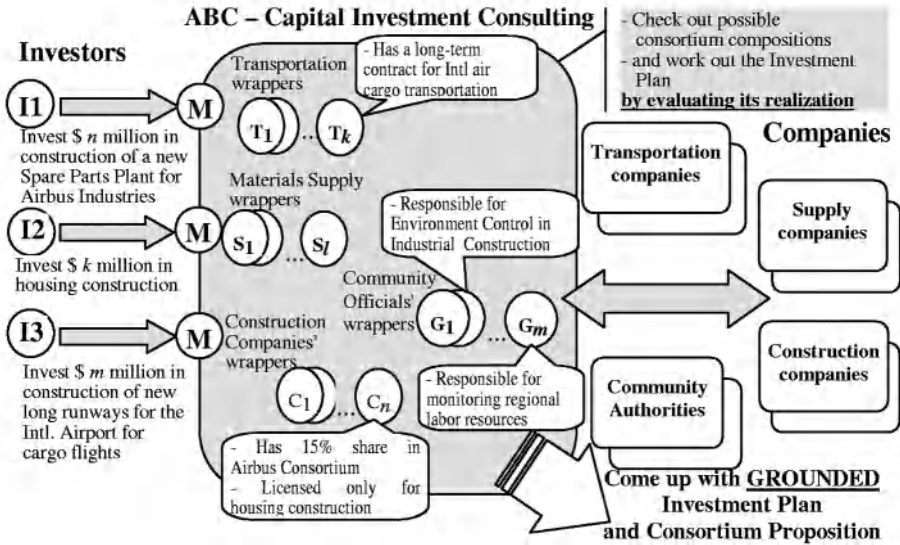


Fig. 1. ABC organization, actors' interests and investors' intentions.

Human investors and their assistant agents **I1**, **I2**, **I3** provide external influences in the form of tasks "Work out investment proposal ...". ABC organization is modelled by Multi-Agent System (MAS), comprising executive agents with specialisations: **M**, **T**, **S**, **G**, **C**. The combination of their commitments, goals and intentions forms the degree of uncertainty, that affects the execution of the tasks provided by the investors and may result in success or failure of the task execution. Actor instances are modelled by software agents designed in frame of the generic formal approach [20]. Individual behaviours of the certain actors $T_1, \dots, T_k, \dots, C_n$ are generally the functions of the four parameters: *capability*, *capacity*, results *desirability*, *credibility* (see Section 4).

The following simple example will further on be used to illustrate the use of Task and Negotiation Ontologies in the process of Investment Plan Evaluation.

Suppose a perspective human investor coming up to a User Assistant agent **I3** to generate the proposition "Work out the investment proposal for the investment of USD 300 million in construction of new long runways for the Intl. Airport to enhance cargo flight reception capacities. The result is required within 1 week starting from now." The formulation of the request is guided by **I3**'s Investment Domain Ontology (not discussed in this paper) as shown on Fig. 2.

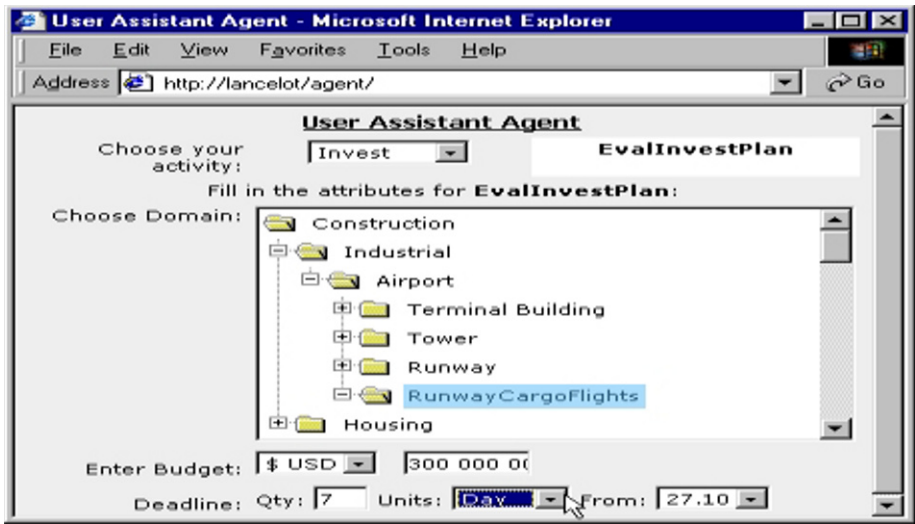


Fig. 2. I3 assisting in the formulation of "Work out investment proposal ..." task

ABC Project Manager agent (**M**) receives this proposition from **I3** as a task in a formalized way as shown on Fig. 3. This task is further analyzed, decomposed and collaboratively executed by the dynamic coalition of ABC member actors. The guidance on how to decompose the task and who are capable to perform its parts is distributed among the knowledge bases of the participant agents and is formalized by *Task Model* (Section 4) and *Task Ontology* (Section 6). Agents join this task coalition by participating in Negotiation on Activity Placement. The pattern for these negotiations is provided in the frame of the Arrangement Phase (Section 5). The semantics of the knowledge the agents are exchanging with each other within the Arrangement Phase is formalized by *Negotiation Ontology* (Section 6).

4 Task Model

It is assumed that a *task* $T = \{w^1, w^2, \dots, w^k\}$ is the set of one or more *activities* (Ref. Eg. [7]). Each activity w , being atomic for a given actor, may be recognised as a task by another actors according to their knowledge about the incoming influence. EvalInvestPlan activity **I3** passes to **M** in our example is decomposed by **M** into the task⁴:

$$T = \{ w^1 = (\text{PerformArchitecturalDesign}, \dots), \\ w^2 = (\text{GetAirTrafficControlPermission}, \dots), \\ w^3 = (\text{CheckEnvironmentalRegulations}, \dots), \\ w^4 = (\text{BuildRunway}, \dots), \\ \dots, \\ w^k = (\text{AssembleInvestPlan}, \dots), \}.$$

⁴ The details are omitted for brevity.

Actors are also capable to generate activities without any external influence in response to some events or in the course of activity performance. The semantics of a task is structured according to the *Task Ontology*. The ontology is used for:

- Checking if the incoming task contains activities known to the actor
- Decomposing the activities which are recognized as being non-atomic
- Reasoning if the parameters and the results' templates ([20]) match to the actor's knowledge about the activity
- Reasoning on whether to perform this activity itself or to negotiate on its delegation to its fellow actors

```
(ask-one
:sender          "I3"
:receiver        "M"
:in-reply-to     Null
:reply-with      Null
:language        (XML)
:ontology        (Task)
:contents        (
<task>
  <activity>
    <name>EvalInvestPlan</name>
    <descr>Work out investment proposal: USD 3 000 million,
          construction of a long runway for cargo flights.
          Deadline: 1 week from today.</descr>
    <budget>   <value>USD, 300 000 000</value>
               <format>money</format> </budget>
    <deadline> <value>day, 7, today</value>
               <format>deadline</format> </deadline>
    <param>
<name>Domain</name><value>RunwayCargoFlights</value>
               <format>Category</format> </param>
    <result-template>
               <name>InvestPlan</name>
               <spec>Investplan.dot</spec>
               <format>MSWordDotFile</format>
    </result-template>
    <result-template>
               <name>ConsortProp</name>
               <spec>ConsortProp.dot</spec>
               <format>MSWordDotFile</format>
    </result-template>
  </activity>
</task>
) )
```

Fig. 3. KQML [21] message containing EvalInvestPlan activity specification in XML

Each of the actors involved in task execution has its own beliefs on how, in what sequence, to perform atomic activities and how much effort should be spent to accomplish the activity, provided that it possesses certain working *capacity* related to this certain atomic activity. These beliefs form their subjective Partial Local Plans (PLP) [5]. PLPs are formalised by the Task Ontology and are used by actor's macromodel programs [20] in the course of activity performance. PLP differ from, say, GPGP [22] by the fact they do not contain the subjective beliefs on what would be the actions of the fellow actors. Alternatively, an actor uses its Fellows'

Capabilities Matrix [23] and Fellows' **Credibility** Matrix [5] to reason if one or another fellow actor is possibly capable to perform the activity or is trustworthy, respectively.

In case the overall task is thought in the form of a graph (see for instance [24]) the final shape of this graph may alter depending on the sequence of actors' involvement. The actors are involved into the task execution by the results of negotiation on placing of one or another activity. This negotiation occurs within the Arrangement Phase each time before the activity is delegated to the executive. As far as the actors are allowed to generate activities, a task graph may evidently contain cycles, since a node (an actor) may generate a subtask containing or leading to the performance of the activity, which may finally be assigned to this very node.

After a task is perceived by an actor and possibly decomposed according to its knowledge the actor may:

- Accept and perform some of the atomic activities contained within the task
- Decline some of the activities
- Decide to delegate some of the atomic activities to one its fellows
- Require the performance of some new atomic activities, the execution of which is essential to successful completion of the overall task

Since the actor has not declined one or more of the atomic activities it becomes linked to the process of the task execution and joins the task **coalition**. Task coalition is thus a dynamic open system pending Scott's definition [4].

By joining the coalition an actor pledges to follow some system rules, which regulate the proportion of its benevolence and self-interest. These rules may be classified, following Jennings Commitment-Convention hypothesis [24] as actor's Individual and Joint Commitments and Coalition Conventions:

Rule 1: Relative co-operation commitment. Coalition members are **relatively** committed to co-operatively achieve the overall goal: to accomplish the task with maximally achievable effectiveness (maximal quality, balanced load, minimal time, ...). The ratio of this commitment depends upon the discrepancy between the actor's autonomous intentions and the overall goal of the task coalition.

Rule 2: Activity arrangement convention. Within the Arrangement Phase the coalition member proposing the activity (the Initiator) pledges to truthfully advertise **desirability** function related to the proposed activity. In response, perspective contractors (the Participants) are committed to truthfully report about their readiness to perform the activity providing the information about their **capacity** share by replying with the duration of activity execution in the form of parametric feedback [20].

Rule 3: Results delivery commitment. Since an activity is accepted by the actor for the performance the actor pledges to unconditionally accomplish this activity and to bring up the results to public immediately after the work is done.

It is assumed that the actor A is characterized by its **capacity** $N_A(w^j)$ with respect to the certain activity w^j in a discrete time space. Capacity is understood as actor's ability to accomplish the activity per unit time interval τ . If, for instance, A is delivering construction materials to the cite, than $N_A(w^j = ("deliver\_1\_tone\_of\_concrete", ...)) = 4$ in case A has 1 ready-mix truck able to deliver up to 4 tones of concrete per τ and will be doubled if A gets one more


```

<Desirability>
  <Activity> <Name>DeliverConcrete</Name> </Activity>
  <Deadline> <Value>27.10.2001/20.00</Value>
    <Format>datetime</Format> </Deadline>
  <Time>
    <ZeroPoint>
      <Value>27.10.2001/08.00</Value>
      <Format>datetime</Format></ZeroPoint>
    <Granularity><Value>2</Value>
    <Format>hours</Format></Granularity>
  </Time>
  <PointsNo>6</PointsNo>
  <TdfPoint>
    <TimeIncr>0</TimeIncr>
    <Incentive><Value>300</Value>
    <Format>Money</Format></Incentive> </TdfPoint>
  ...
  <TdfPoint>
    <TimeIncr>5</TimeIncr>
    <Incentive><Value>0</Value>
    <Format>Money</Format></Incentive> </TdfPoint>
</Desirability>

```

Fig. 4. Desirability function (XML) for DeliverConcrete activity results

carrier of the same type. Actor's capacity may be considered **unlimited** — as if at any time it is able to deliver as much concrete as needed. Note that in this case w^j accomplishment duration will still be exactly $1 \times \tau$ as far as the actor operates with the certain type of trucks and minimal duration may not be less than one unit time interval. This ideal case is the case with no activities waiting in line. In case the actors' capacity is **limited** (as if it has the fixed number of ready-mix trucks at its disposal) $N_A(w^j)$ may be evenly, or not really evenly, say, according to the priority the customer has, distributed over the activities to be performed.

Activity w^j may be constrained by the **deadline** d_{w^j} . The deadline is the point in time after which w^j results are not needed anymore by the customer agent. For example, the construction company will not need any concrete on Sunday instead of Friday. This means that w^j results **desirability** function value (see *TÆMS* quality property [22] for the kind of an indirect analogy):

$$des_{w^j}(t, d_{w^j}) = \begin{cases} tdf(t), t \leq d_{w^j} \\ 0, t > d_{w^j} \end{cases} \quad (1)$$

falls down to zero after the deadline has passed and promises changing incentive $tdf(t)$ as a kind of a trade-off over time. For the concrete delivery example this desirability may be as shown on Fig. 4.

5 Arrangement Phase

The role of the Arrangement Phase is to seek for the executor of the certain activity, which is decided (according to PLP or because of the overload) to be delegated to another actor. Arrangement is performed via negotiation between the Initiator and the group of Participant actors. Initiator's goal is to perform the activity in the most optimal way. It is assumed that in the course of negotiation on activity placement Initiator actor attempts to solve two-criteria optimisation problem. First criterium is related to the believed optimal time of the activity accomplishment. Second one is the

Arrangement Phase Protocol

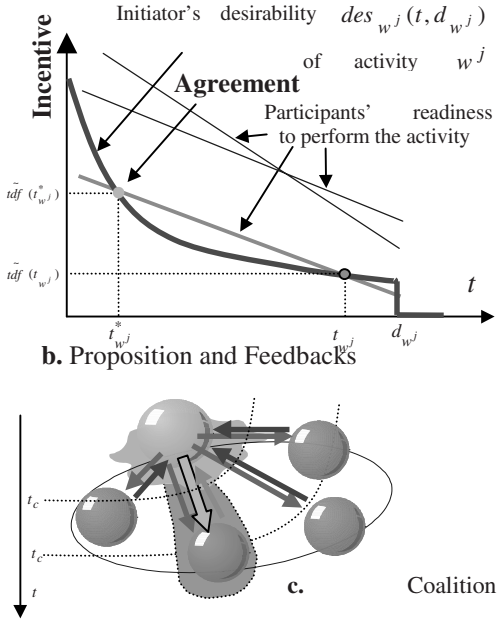
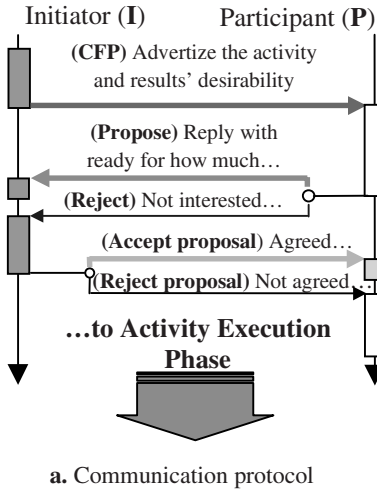


Fig. 5. Arrangement phase. Negotiation on delegating the activity and joining the coalition

optimal incentive to be paid to the fellows. The proposed incentive values are derived from the activity budget (Task Ontology). The protocol used for peer-to-peer communication is a kind of FIPA CNP [25] (Fig. 5a) Initiator multicasts activity results' desirability function (1) to negotiation Participants. The Participants reply with their 2-point trade-off assessment feedbacks indicating their readiness to perform the proposed activity (Fig. 5b). The participant is ready to perform the activity in case it's feedback contains intersections with the desirability function declared by the Initiator. Otherwise, it rejects the proposition. Initiator than chooses the best bid from the set of received intersections. The participant, which feedback has been chosen as the optimal bid, becomes the contractor. It thus joins the task coalition (Fig. 5c) and proceeds with delegated activity performance. More details on the behaviour models of the parties within the Arrangement Phase may be found in [23, 26].

Let's observe which shared concepts (marked bold) are used by ABC actors while arranging the performance of DeliverConcrete activity. C_i becomes the Initiator of the Arrangement Phase after it has generated DeliverConcrete **activity** in the course of BuildRunway **task** decomposition, has realised (**PLP**) that it is not capable to perform this activity itself and has detected that $T_1, \dots, T_n$ are the proper candidates for the job according to it's Fellows' Capabilities Matrix. C_i multicasts DeliverConcrete activity **specification** and results **desirability** function to $T_1, \dots, T_n$. $T_1, \dots, T_n$ in turn benevolently become the Participants after perceiving this influence from C_i and report their attitudes to the proposition in the form of 2-point trade-off assessment feedbacks (Fig. 6). These feedbacks are based on the assessment of the capacity share or the **effort** the participant is ready to spend to the activity performance.

```

<TdfFeedback>
  <activity>          <name>DeliverConcrete</name>          </activity>
  <PointsNo>2</PointsNo>
  <TdfPoint>
    <TimeIncr> $t_{w^j}^*$ </TimeIncr>    <Incentive><value> $\tilde{tdf}(t_{w^j}^*)$ </value>
                                     <format>Money</format></Incentive>
  </TdfPoint>
  <TdfPoint>
    <TimeIncr> $t_{w^j}$ </TimeIncr>    <Incentive><value> $\tilde{tdf}(t_{w^j})$ </value>
                                     <format>Money</format></Incentive>
  </TdfPoint>
</TdfFeedback>

```

Fig. 6. Example of Trade-off Assessment Feedback (XML) for DeliverConcrete activity

6 Task and Negotiation Ontologies

Task and Negotiation ontologies were designed and formalized in Standard OIL [27] to serve as shared conceptualisations in frame of the reported approach to model collaborative task execution by coalitions of service providing agents. The framework and the design of these ontologies was primarily motivated by B2B E-Commerce features of uncertainty and the need of the proper balance between self-interest, rationality and benevolence. The role of the Task Ontology is to provide shared concepts of a task, an activity, a parameter, a result template an effort, a priority, a deadline, a budget as well as a Partial Local Plan for activity performance. These concepts are used by service providing agents to determine: if the incoming activity is atomic; if they are capable to perform the activity; if the parameters and the expected results match to their knowledge about the activity; if they need their fellows' assistance to perform the activity. ER-style diagram of the Task Ontology is given on Fig. 7.

Negotiation Ontology provides shared conceptualization of the terms used by agents while they participate in negotiations on activity placement. Negotiation type for the Arrangement Phase is Contracting. The difference of the negotiation approach used in the reported research from another contracting frameworks is the use of parametric trade-off assessment feedbacks. This parametrization provides more flexibility to agents' behaviour and allows to avoid negotiation iterations. The shared concepts of Negotiation Ontology are: activity, results' desirability, proposed deadline, time and its granularity, incentive, trade-off point, trade-off feedback. ER-style diagram of the Negotiation Ontology is presented on Fig. 8.

OilEd 2.2a⁵ and FACT⁶ reasoner were used for ontologies design and expressiveness check respectively. OIL, RDFS, DAML and SHIQ versions of the reported Task and Negotiation Ontologies are available at http://eva.zsu.zp.ua/eva_personal/ontologies/.

⁵ Freely downloadable from <http://img.cs.man.ac.uk/oil/>. Last accessed on Nov. 3, 2001.

⁶ <http://www.cs.man.ac.uk/~horrocks/FaCT/>. Last accessed on Nov. 3, 2001.

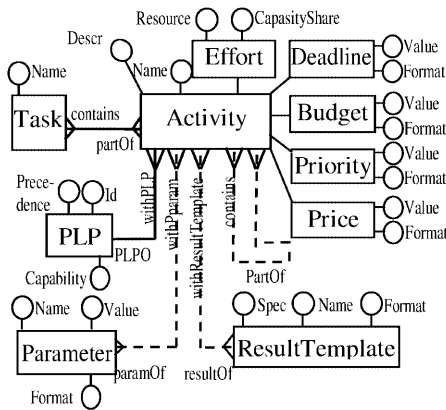


Fig. 7. Er-style diagram of Task Ontology

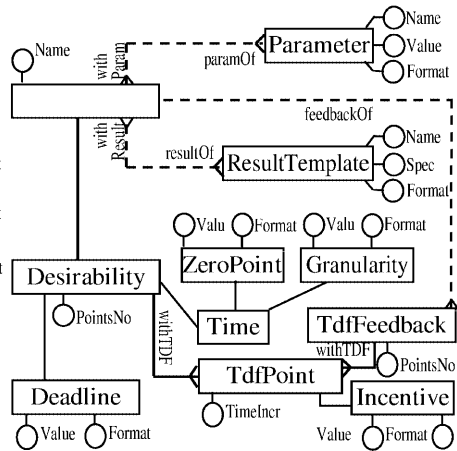


Fig. 8. ER-style diagram of Negotiation Ontology

7 Conclusions and Future Work

Co-ordination patterns for distributed activity performance in E-Commerce domain should take into account the balance of the actors' features of rationality, self-interest and benevolence. In frame of the presented research in progress this goal is approached by the ability of the actors to dynamically form the coalitions for optimal task execution. Coalition formation is co-ordinated by contracting negotiation on one or another activity placement. The role of the Task and Negotiation Ontologies presented in the paper is to provide the shared conceptualisation of the terms, the structures and the procedures used by agents in the processes of activity analysis, decomposition, performance and delegation. The ontologies are formalized in OIL and are translated to DAML (RDF), RDFS, SHIQ notations, thus providing the concepts in the forms of emerging service mark-up standards in frame of, eg., W3C activities.

The bulk of the planned future work contains: modelling and conceptualisations of alternative negotiation patterns (iterative bargaining, auction, ...); further enhancement of the Task Ontology by providing means for Problem Solving Method and Macromodel Procedure description (this may require the development of OIL extensions); development of a prototype agent-based application for B2B Investment Consulting Marketplace.

Acknowledgements. Presented research is run in frame of the project funded by Ukrainian Ministry of Science and Education (Grant No 0199Y1571). The authors would like to express their special gratitude to Hans Akkermans, Dieter Fensel and other colleagues from Business Informatics research group of Free University of Amsterdam for fruitful discussions on ontologies, semantics and OIL in E-Commerce domain.

References

1. Berners-Lee, T., Fischetti, M.: *Weaving the Web: The original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper, San Francisco (1999)
2. McIlraith, S. A., Son, T.C., Zeng, H.: *Mobilizing the Semantic Web with DAML-Enabled Web Services*. In: *Proc. Of the Second Intl. Workshop on the Semantic Web – SemWeb'2001*, Honkong, China, May 1, 2001 (2001) 82–87
3. Nwana, H. S.: *Software Agents: an Overview*. *Knowledge Engineering Review* 11(3) (1996) 205–244
4. Richard Scott, W.: *Organizations: Rational, Natural and Open Systems*. Prentice-Hall, Inc. Englewood Cliffs, NJ (1987)
5. Ermolayev, V. A., Borue, S. U., Tolok, V. A., Keberle, N. G., Plaksin, S. L.: *Arranging Co-operative Activities in Dynamic Coalitions of Self-Interested Actors*. Technical Report. Dept. of Math. Modelling and IT, Zaporozhye State University (2001)
6. Tate, A.: *Roots of SPAR - Shared Planning and Activity Representation*. *Knowledge Engineering Review* 13(1) (1998) 121–128
7. Uschold, M., King, M., Moralee, S., Zorgios, Y.: *The Enterprise Ontology*. *Knowledge Engineering Review* 13(1) (1998)
8. Schlenoff, C.; Ciociu, M.; Libes, D.; Gruninger, M.: *Process Specification Language: Results of the First Pilot Implementation*. In: *Proc. of the Int. Mechanical Engineering Congress and Exposition*, Nashville (1999)
9. Fox, M.C.; Gruninger, M.: *Enterprise Modelling*. *AI Magazine* 19(3) (1998) 109–121
10. Gruninger, M., Fox M. C.: *An Activity Ontology for Enterprise Modelling*. *Workshop on Enabling Technologies – Infrastructures for Collaborative Enterprises*, West Virginia Univ. (1994)
11. Tate, A.: *<I-N-OVA> and <I-N-CA> - Representing Plans and other Synthesized Artifacts as a Set of Constraints*. In: *Proc. of the AAAI-2000 Workshop on Representational Issues for Real-World Planning Systems*, Austin (2000)
12. Faratin, P.: *Automated Service Negotiation Between Autonomous Computational Agents*. PhD Thesis, Univ. of London (2000)
13. Tsvetovat, M., Sycara, K., Chen, Y., Ying, J.: *Customer Coalitions in the Electronic Marketplace*. In: *Proc. of Agents 2000 Conference*, Barcelona, Spain (2000)
14. Faratin, P., Jennings, N. R., Buckle, P., Sierra, C.: *Automated Negotiation for Provisioning Virtual Private Networks Using FIPA-Compliant Agents*. In: *Proc. 5th Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM-2000)*, Manchester, UK (2000) 185–202
15. Jennings, N. R., Faratin, P., Norman, T. J., O'Brien, P., Odgers, B., Alty, J. L.: *Implementing a Business Process Management System using ADEPT: A Real-World Case Study*. *Journal of Applied Artificial Intelligence* 14 (5) (2000) 421–465
16. Barbuceanu, M.; Fox, M.S.: *The Architecture of an Agent Building Shell*. In: (Wooldridge, M.; Mueller, J.P.; Tambe, M. Eds): *Intelligent Agents II: Agent Theories, Architectures and Languages*. *Lecture Notes in Artificial Intelligence*, Vol. 1037, Springer-Verlag, Berlin Heidelberg New York (1996)
17. Sycara, K., Klusch, M., Widoff, S., Lu, J.: *Dynamic Service Matchmaking Among Agents in Open Information Environments*. *ACM SIGMOD Record* 28(1) Special Issue on Semantic Interoperability in Global Information Systems (1999) 47–53
18. Borue, S. U., Ermolayev, V. A., Tolok, V. A.: *Application of Diakoptical MAS Framework to Planning Process Modelling*. *Problems of Programming. The Scientific Journal* 1-2, 2000. Special Issue: *Proc. of the 2-nd Int. Scientific - Practical Conference on Programming (UkrPROG'2000)*, Kiev, 23-26 May 2000 (2000) 488–500
19. Ermolayev, V., *Dynamic Agent Communities Facilitating to Distant Learning in a Virtual University Information Space*. In: *Proc. of Intl. Conf. IS2000, Special Session on Virtual Universities and Distance Education*, Japan, November 5-8, 2000 (2000) 488–495

20. Ermolayev, V. A.; Borue, S. U.; Tolok, V. A.; Keberle, N. G.: Use of Diakoptics and Finite Automata for Modelling Virtual Information Space Agent Societies. Lecture Notes of Zaporozhye State University, 3(1) (2000) 34–44
21. Labrou, Y., Finin, T. A Proposal for a New KQML Specification. TR CS-97-03, Univ. of Maryland Baltimore County, Feb. 1997 (1997)
22. Decker, K.S.: Environment Centered Analysis and Design of Coordination Mechanisms. Umass CMPSCI Tech. Rep. 95-69, (1995)
23. Ermolayev, V. A, Tolok, V. A.: Modelling Distant Learning Activities by Dynamic Agent Task Coalitions. To appeare In: (Noguchi, S., Yu, C. Eds.): Enabling Society with Information Technology. Springer-Verlag, Tokyo (2001)
24. Jennings, N. R.: Coordination Techniques for Distributed Artificial Intelligence. In: (O'Hare, G. M. P., Jennings, N. R. Eds.) Foundations of Distributed Artificial Intelligence. Wiley (1996)
25. Foundation for Intelligent Physical Agents. FIPA Contract Net Interaction Protocol Specification. Version E. Ref. No XC00029E. (2001)
26. Ermolayev, V. A., Plaksin, S. L.: Co-ordination of Work Placement in Agents' Task Coalitions. Lecture Notes of Zaporozhye State University, 4(2), to appear (2001)
27. Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., Van Harmelen, F., Klein, M., Staab, S., Studer, R.: OIL: The Ontology Inference Layer, Tech.Rep. IR-479, Vrije Universiteit Amsterdam, Sept., 2000 (2000)

A Multi-perspective Methodology for Modelling Inter-enterprise Business Processes

Kuldar Taveter^{1,2} and Gerd Wagner³

¹ VTT Information Technology (Technical Research Centre of Finland),
P.O.Box 1201, FIN-02044 VTT, Finland
Kuldar.Taveter@vtt.fi

² Tallinn Technical University, Department of Informatics,
Raja 15, 12618 Tallinn, Estonia

³ Eindhoven University of Technology, Faculty of Technology Management,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands,
G.Wagner@tm.tue.nl
<http://tmitwww.tm.tue.nl/staff/gwagner>

Abstract. We propose an agent-oriented methodology for modelling business processes between enterprises that consists of the steps of analysis and design. The analysis starts by modelling intentional dependencies between the actors of the problem domain at hand, and continues by modelling intentional relationships that are internal to the actors, such as task decomposition links. In the design, intentional dependencies between the actors are transformed to commitment-based models of interactions between the agents. Models of tasks performed by the actors, obtained as a result of means-ends analysis, are transformed to activities and reaction rules, defining the behaviours of agents. The methodology is evaluated by using the case study of an electronic advertising process in newspapers.

1 Introduction

In information systems engineering, there is a gap between descriptive models needed at the stage of analyzing the problem domain at hand and prescriptive models needed later on at the stage of designing the information system for the problem domain. The gap is there because the notions used in modelling problem domains and information systems are too different. The new emerging paradigm of *agent-orientation* helps to bridge this gap by enabling to use the same terms like *beliefs*, *perceptions*, *memory*, *goals*, *commitments*, etc. for humans, institutions, and artificial (e.g., software-controlled) systems. According to this paradigm, communication between different artificial (software-controlled) systems and between systems, institutions, and humans is understood as communication between agents. Since the paradigm of agent-orientation promotes *autonomous action and decision-making*, it is highly relevant for modelling and implementing business processes [16, 20] involving different enterprises with their respective information systems.

2 The Methodology of Agent-Oriented Modelling

2.1 The Modelling Process

We propose an agent-oriented methodology for modelling business processes between enterprises that consists of the steps of analysis and design. For the step of analysis, we make use of the *i** modelling framework proposed in [21, 25]. The analysis starts by modelling intentional dependencies between the actors of the problem domain at hand, and continues by modelling intentional relationships that are internal to the actors, such as task decomposition links.

For the step of design, we use the Agent-Object-Relationship (AOR) modelling proposed in [2, 8]. In the design, intentional dependencies between the actors are transformed to commitment-based models of interactions between the agents. Models of tasks performed by the actors, obtained as a result of means-ends analysis, are transformed to models of activities and reaction rules, defining the behaviours of agents.

2.2 Analysis of the Problem Domain by *i**

For the analysis of the problem domain at hand, we make use of the *i** framework proposed in [21, 25]. The *i** (which stands for “distributed intentionality”) framework provides understanding of the motivation of social actors that depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. The framework consists of a **Strategic Dependency (SD)** model and a **Strategic Rationale (SR)** model. The SD model provides an intentional description of a (business) process in terms of a network of dependency relationships among actors. The SR model provides an intentional description of a (business) process in terms of process elements and the rationales behind them [21].

Analysis of Dependencies. The SD model consists of a set of nodes and links. Each node represents an actor, and each link between two actors indicates that one actor depends on the other for something in order that the former may attain some goal. The depending actor is called the **depender**, and the actor who is depended upon the **dependee**. The object around which the dependency relationship centres is called the **dependum**. An **actor** is an active entity that carries out actions to achieve goals by exercising its knowhow. The term *actor* is used to refer generically to any unit to which intentional dependencies can be ascribed.

Four types of dependencies are distinguished among actors, based on the type of dependum. In a **goal dependency**, a depender depends on the dependee to bring about a certain state in the world. The dependee is given the freedom to choose how to do it. In a **task dependency**, a depender depends on the dependee to carry out an activity. A task dependency specifies *how* the task is to be performed, but not *why*. In a **resource dependency**, the depender depends on the dependee for the availability of an entity (physical or informational). By establishing this dependency, the depender gains the ability to use this entity as a resource. In a **softgoal dependency**, a depender depends

on the dependee to perform some task that meets a *softgoal*. The meaning of the softgoal is not clear-cut. It is specified in terms of the methods that are chosen in the course of pursuing the goal.

The intentional dependencies of the domain of newspaper advertising, distilled from [12], are represented in Figure 1. The figure includes five actor types: the *Advertiser*, *Media Agency*, *Publication*, *Artwork Designer*, and *Artwork Producer*. The *Advertiser* depends on the *Media Agency* to have an advertising campaign performed. The *Advertiser* is only concerned about the outcome of the campaign and doesn't care *how* exactly the campaign is organized by the *Media Agency*. Therefore the dependency “*Campaign performed*” would be appropriately modelled as a goal dependency.

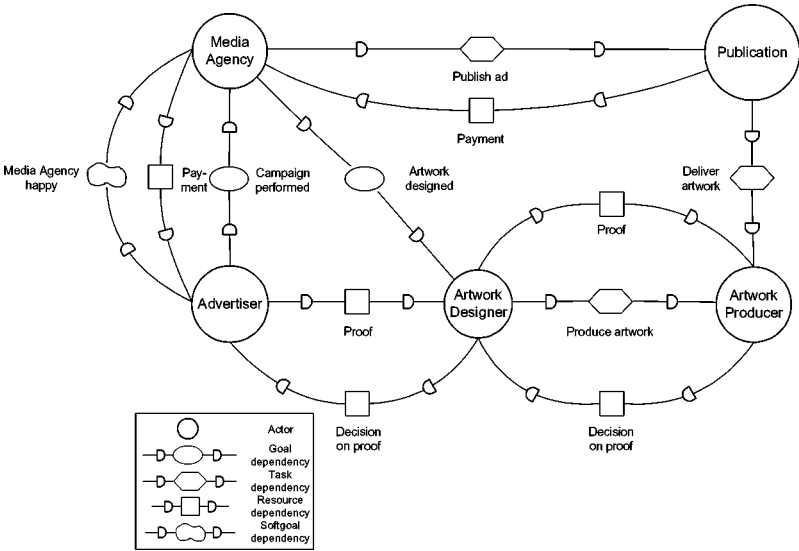


Fig. 1. The Strategic Dependency Model for the domain of newspaper advertising

Another goal dependency in Figure 1 is that the *Media Agency* depends on the *Artwork Designer* to have an artwork designed. Again, the *Media Agency* wants to have the artwork designed and doesn't care *how* it is done. Moreover, due to the artistic nature of designing an artwork, it is not even possible to provide the *Artwork Designer* with precise instructions how the artwork should be designed. As an example of a task dependency, the *Media Agency* depends on the *Publication* to have the ad published. Since the *Media Agency* wants to have the ad published according to the well-defined instructions established by it, the dependency “*Publish ad*” would be appropriately modelled as a task dependency. Another example of a task dependency is the dependency “*Produce artwork*” between the *Artwork Designer* and *Artwork Producer* where the *Artwork Designer* wants to have the artwork produced exactly according to the instructions provided by it. The dependency “*Deliver artwork*” between the *Publication* and *Artwork Producer* is naturally also modelled as a task dependency.

The dependencies of the *Media Agency* and *Publication* on respectively the *Advertiser* and *Media Agency* for a payment are modelled as resource dependencies where the dependum is an amount of money.

The *Advertiser* depends on the *Artwork Designer* and the *Artwork Designer* on the *Artwork Producer* for a proof of the artwork to be designed or produced. The *Artwork Designer* and *Artwork Producer*, in turn, depend on the *Advertiser* and *Artwork Designer*, respectively, for a decision on the proof. These dependencies are expressed as resource dependencies. The dependums here are proofs and decisions on the proofs.

Since the *Media Agency* acts as a customer for the *Publication*, there is a softgoal dependency “*Media Agency happy*” between the *Publication* and *Media Agency*.

Means-Ends Analysis. The SR model provides an intentional description of a (business) process in terms of process elements and the rationales behind them. While the SD model represents only external relationships between actors, the corresponding SR model describes intentional relationships that are internal to actors, such as task decomposition links.

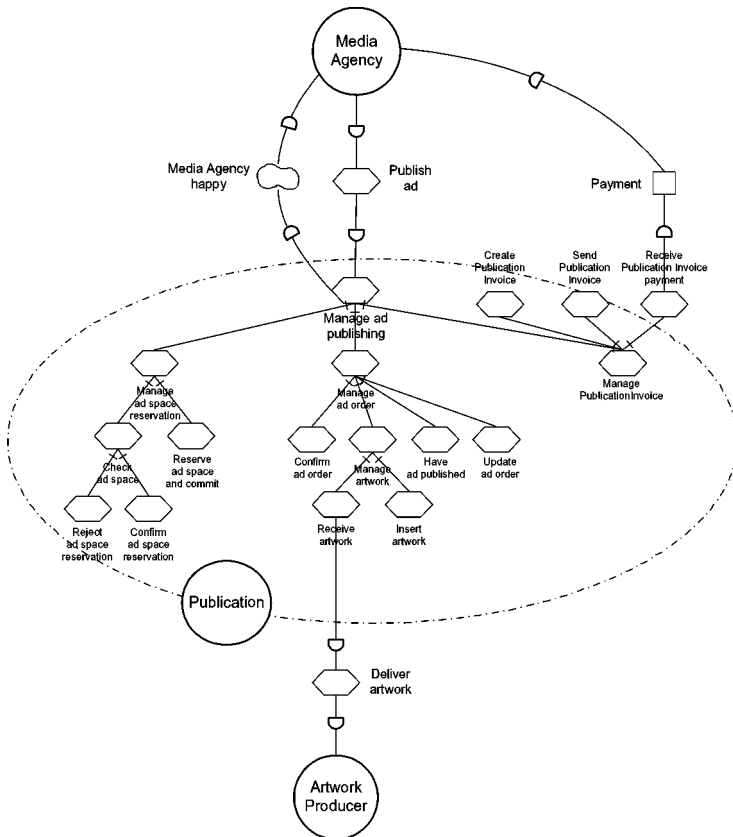


Fig. 2. The Strategic Rationale Model for the *Publication*

The SR model is a graph, with several types of nodes and links. There are four types of nodes, based on the same distinction made for dependum types in the SD model – goal, task, resource, and softgoal. A **goal** is a condition or state of affairs in the world that the actor would like to achieve. A **task** specifies a particular way of doing something. A **resource** is a physical or informational entity to be used or produced by the actor. A **softgoal** is a condition in the world which the actor would like to achieve, but the criteria for the condition being achieved are not sharply defined *a priori*, and are subject to interpretation.

Only tasks are present in Figure 2. A task node is linked to its component nodes by **task decomposition links**. Tasks can also connect up with dependency links in SD model(s), when the reasoning goes beyond an actor's boundary.

The SR model for the *Publication*, created based on [12], is represented in Figure 2. The model shows that the *Publication* is able to fulfill the task dependency “*Publish ad*” that the *Media Agency* depends on by running the internal task “*Manage ad publishing*”. The task “*Manage ad publishing*” consists of the subtasks “*Manage ad space reservation*”, “*Manage ad order*”, and “*Manage Publication Invoice*”. These subtasks, in turn, consist of other subtasks. Some of the subtasks have alternative subtasks like “*Reject ad space reservation*” and “*Confirm ad space reservation*” of the subtask “*Check ad space*”, but their alternativeness cannot be represented by means of i^* . Therefore it is deferred to section 2.3 where it is done by reaction rules of agents. The task “*Manage ad publishing*” and the subtasks “*Receive artwork*” and “*Receive Publication Invoice payment*” respectively connect up with the softgoal dependency “*Media Agency happy*”, task dependency “*Deliver artwork*”, and resource dependency “*Payment*”.

2.3 Design by Agent-Object-Relationship Modelling

Since intentional dependencies between actors of i^* imply inter-agent interactions and process elements internal to actors such as tasks and goals imply activities of agents, we propose to model interactions and activities by Agent-Object-Relationship (AOR) Modelling for designing an inter-enterprise information system. In this section, we will first present an overview of AOR Modelling and will then describe mapping from the SD and SR models of i^* , which we used at the stage of analyzing the problem domain, to the modelling abstractions of AOR modelling.

Principles of Agent-Object-Relationship Modelling. Agent-Object-Relationship diagrams were proposed in [2, 8] as an agent-oriented extension of Entity-Relationship-diagrams, or UML-style class diagrams. In order to capture more semantics of the dynamic and deontic aspects of organizations involved in business processes and their information systems, such as the events and actions related to the ongoing business processes, it is proposed to make an ontological distinction between active and passive entities, that is, between *agents* and *ordinary objects*. AOR modelling suggests that the semantics of business transactions can be more adequately captured if the specific *business agents* associated with the involved events and actions are explicitly represented in organizational as well as in inter-organizational information systems in addition to passive *business objects*.

Entity types. In AOR modelling, an entity is either an *agent*, an *event*, an *action*, a *claim*, a *commitment*, or an ordinary *object*. An organization is viewed as a complex **institutional agent**, being involved in a number of interactions with external agents and consisting of a number of internal agents that perceive events and perform actions on behalf of it. **Internal agents** may be humans, artificial agents (such as software agents, agentified information systems, robots, or agentified embedded systems), or institutional agents (such as organization units).

The entity types listed above can be viewed and represented as the corresponding stereotypes of UML [28]: <<agent>>, <<event>>, <<action>>, <<activity>>, <<claim>>, <<commitment>>, and <<object>>.

As usual, entity types are visually represented by rectangles while relationship types are represented by connection lines (possibly with crows feet endings in order to indicate multiplicity). While an *object type* is visualized as an ordinary rectangle, an *agent type* is graphically rendered as a rectangle with rounded corners. An internal agent type is visualized by such a rectangle with a dashed line drawn within the institutional agent rectangle it belongs to (like Department in Figure 3). An instance of an agent type is distinguished from an agent type by underlining its name (like the Publication Secretary in Figure 3).

Actions and events. In a business domain, there are various types of actions performed by agents, and there are various types of state changes, including the progression of time, that occur in the environment of the agents. For an external observer, both actions and environmental state changes constitute events. In the internal perspective of an agent that acts in the business domain, only the actions of other agents count as events.

Actions create events, but not all events are created by actions. Those events that are created by actions, such as delivering a product to a customer, are called **action events**. Examples of business events that are not created by actions are the fall of a particular stock value below a certain threshold, or a timeout in an auction. Actions make up activities. While an action happens at a time point (i.e., it is immediate), an **activity** is being performed during a time interval (i.e., it has duration), and consists of a set of actions. We distinguish between a **human activity** and an **automated activity** performed by e.g. a software agent.

We make a distinction between *communicative* and *non-communicative* actions and events. Many typical business events, such as receiving a purchase order or a sales quotation, are communication events. The expressions *receiving a message* and *sending a message* may be considered to be synonyms of *perceiving a communication event* and *performing a communication act*.

Business communication may be viewed as **asynchronous** point-to-point message passing. The feature of asynchronous communication is a fundamental one in agent-oriented modelling that makes the latter especially suitable for modelling inter-enterprise business processes. This feature distinguishes agent-oriented modelling from other modelling techniques requiring synchronous communication like e.g. Role Activity Diagrams [29].

As opposed to the low-level (and rather technical) concept of messages in object-oriented programming, AOR modelling assumes the high-level semantics of speech-act-based *Agent Communication Language (ACL)* messages (see [18, 19]).

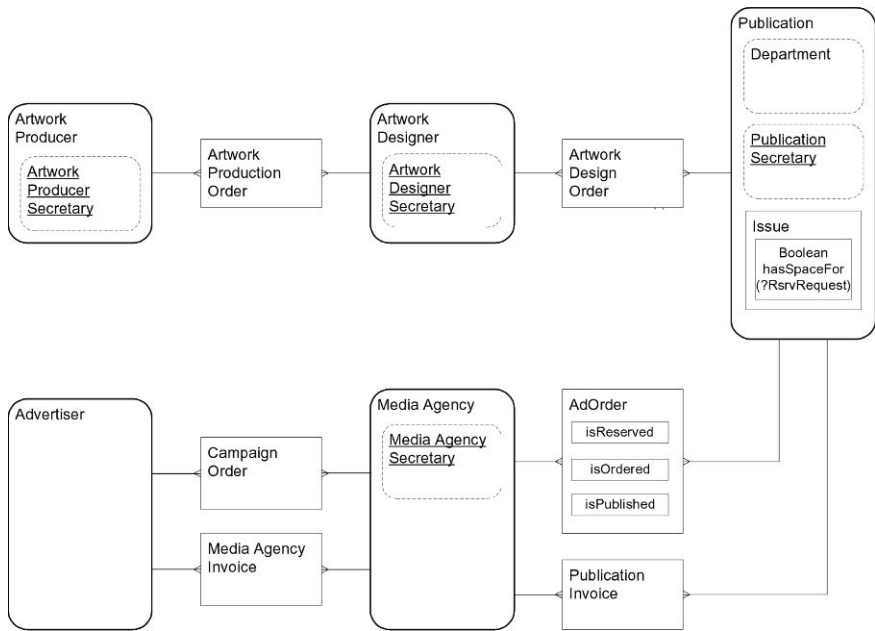


Fig. 3. The agent diagram of the domain of newspaper advertising. The agents include the respective (human) internal agents Media Agency Secretary, Publication Secretary, Artwork Designer Secretary, and Artwork Producer Secretary, and (instances of) the (institutional) internal agent type Department which is not further specified. The agents have knowledge/information about shared objects of certain types. An entity of the class AdOrder can have the status *isReserved*, *isOrdered*, or *isPublished*. The object type Issue has the intensional predicate *hasSpaceFor* (*?RsrvRequest*).

Commitments and claims. Commitments are fundamental components of business interaction processes. Representing and processing commitments and claims in information systems explicitly helps to achieve coherent behaviour in interaction processes. In [26], the social dimension of coherent behaviour is emphasized, and a **social commitment** $\mathbf{C}(x, y, G, p)$ is defined as a relation between the *debtor* x , the *creditor* y , and the *discharge condition* p that the debtor is committed to, in the scope of the *social context* G . The context is a group that *contains* the participating agents and has some sovereignty over its members. In the example of newspaper advertising, the context group imposes certain external constraints on the operation of media agencies and publications, like the ones stating how claims and disputes should be settled between a media agency and a newspaper.

A commitment $c = \mathbf{C}(x, y, G, p)$ is **base-level** if p does not refer to any other commitments; c is a **metacommitment** if p refers to a base-level commitment. Metacommitments create a society where the metacommitments are the norms of this society [26]. In our case study of newspaper advertising, the society consists of agents of the types shown in Figure 1.

Commitments typically arise from certain communication acts. For instance, sending a sales quotation to a customer commits the vendor to reserve adequate stocks of the

quoted item for some time. Likewise, acknowledging a sales order implies the creation of a commitment to deliver the ordered items on or before the specified delivery date.

There are two kinds of commitments: commitments to do an action and commitments to see to it that some condition holds. The former are called *to-do* commitments, and the latter *see-to-it-that* commitments. A *to-do* commitment is discharged by performing the corresponding action, while a *see-to-it-that* commitment is discharged by performing an action or activity whose state changing effect implies the discharge condition.

In the perspective of a particular agent, commitments of other agents are viewed as *claims* against them.

Modelling Business Processes by Interaction Frames. The i^* framework enables to describe and analyze commitments that exist between the actors of a problem domain. We are also interested in how commitments arise and evolve, i.e. in prescriptive models of commitments. According to [21], an additional step needs to be taken to arrive at prescriptive requirements specifications from i^* . We propose to achieve a more prescriptive stance of modelling through representing intentional dependencies as metacommitments by using the notion of social commitment.

For a goal dependency in the i^* framework, the condition p is an assertion *achieved*(g) representing the goal g to be achieved by an agent; for a task dependency on task t , p is *done*(t); for a resource dependency on resource r , p is *avail*(r) [21]. For mapping from i^* to AOR, we represent an intentional dependency as a *metacommitment* of the depender towards the dependee to create (a) commitment(s) to satisfy the condition p . For example, the goal dependency “Campaign performed” in Figure 1 of the Advertiser on the Media Agency can be represented as a metacommitment of the Media Agency towards the Advertiser by which the Media Agency commits upon receiving an order for campaign from the Advertiser to create a commitment towards the Advertiser to achieve the goal “Campaign performed”. We model business processes through lifecycles of commitments, governed by metacommitments, by using commitment protocols [17, 24]. After that we capture lifecycles of commitments by using interaction frames of external AOR models. In an **external AOR model**, we adopt the view of an external observer who is observing the (prototypical) agents and their interactions in the problem domain under consideration. Typically, an external AOR model will have a *focus*, that is an agent, or a group of agents, for which we would like to develop an interaction model. In the view of an external observer, all actions of agents are at the same time also events, and commitments are also claims, exactly like two sides of the same coin. Therefore, an external AOR model contains, besides the agent and object types of interest, the action event types and commitment/claim types that are needed to describe the interaction between the focus agent(s) and the other types of agents.

An **interaction frame diagram**, in an external AOR model, provides a static picture of possible interactions and involvement of commitments/claims within them. The interaction frame diagram, covering the business process type of organizing an advertising campaign, is depicted in Figure 4.

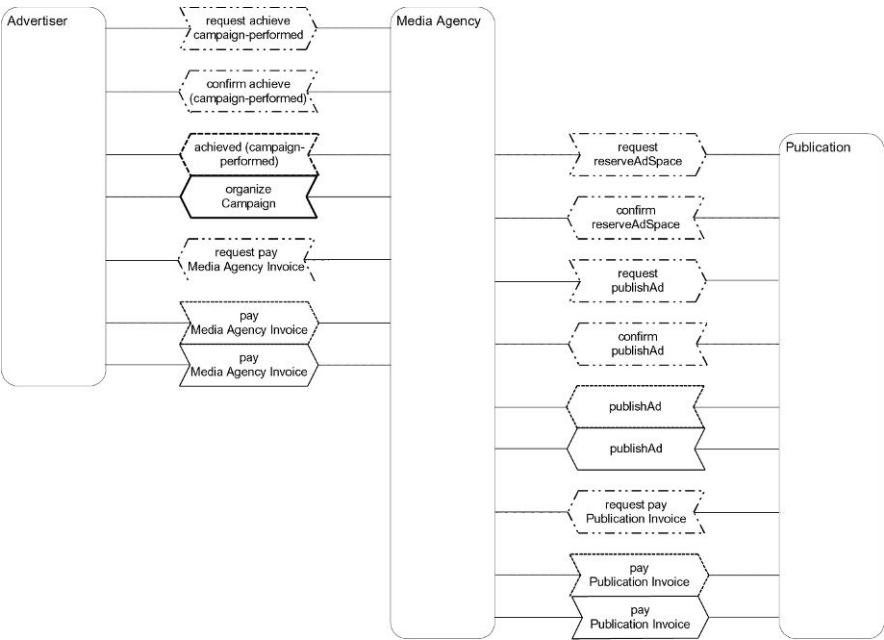


Fig. 4. The interaction frame for the business process type of organizing an advertising campaign with three focus agents. The frame starts with the formation of the *see-to-it-that* commitment achieved(campaign-performed) between agents of the types Advertiser and Media Agency. The occurrence of an action event of the type publishAd between Media Agency and Publication is preceded by the formation and refinement of the corresponding commitment/claim of the type publishAd, as a consequence of the exchange of communicative action events of the types request reserveAdSpace, confirm reserveAdSpace, request publishAd, and confirm publishAd. The occurrence of a communicative action event of the type request pay Publication Invoice gives rise to a commitment/claim of the type pay Publication Invoice which is satisfied by the occurrence of an action event of the type pay Publication Invoice. The latter also concludes the activity organizeCampaign, as its discharge condition achieved (campaign-performed) becomes true. Finally, Media Agency sends a Media Agency Invoice to Advertiser, the latter commits and pays the Media Agency Invoice.

In the graphical notation of AOR modelling, *to-do* commitments are coupled with the action events satisfying them. For example, in Figure 4 the commitment of the publishAd type is coupled with the action event of the corresponding type. A *see-to-it-that* commitment is coupled with the activity completing of which makes the discharge condition *p* of the commitment true. For example, in Figure 4 the *see-to-it-that* commitment achieved(campaign-performed) is satisfied by the activity organizeCampaign which consists of several action events related to having the ad(s) published in one or more publications.

Modelling Agent Behaviours by Activity Diagrams. The behaviour of a Knowledge-Perception-Memory-Commitment agent introduced in [4, 10] is encoded by a set of *reaction rules*. A reaction rule does not have any counterpart in UML [28], but it can be viewed as a construct complementing UML.

Reactive behaviour of agents is the most dominant one in a business domain. Reactive behaviour may also be combined with proactive behaviour, but this is not the topic of the present paper.

Each task identified by means-ends analysis (v. section 2.2) is mapped to one or more activities. An activity can be started in response to an event by means of a reaction rule. Once an activity is started, the agent is in the corresponding *activity state*. An activity can trigger reaction rules or subactivities through the implicit triggering event `startOf(activity)`. Additionally, each activity is associated with another implicit triggering event `endOf(activity)` that can trigger other reaction rules and activities.

An activity may also have a *goal*. For example, the goal of the activity `organizeCampaign` in Figure 4 is `campaign-performed`.

Activities are visualized as rectangles with rounded left and right sides, as shown in Figure 5. The triggering event `startOf(activity)` is represented by an empty circle with the outgoing arrow to the activity or reaction rule triggered by it. The triggering event `endOf(activity)` is modelled by drawing just the outgoing arrow from the activity to either the next activity or the reaction rule triggered by it.

A reaction rule is visualized as shown in the legend for Figure 5. If the composition of an activity is not specified, like in case of the activity “*Have ad published*”, the incoming arrow representing a status condition, the outgoing arrow denoting a status change, and the outgoing connector to an action type are directly connected to the rectangle representing the activity.

In Figure 5, the activity “*Manage ad publishing*” performed by the Publication and its constituent activities “*Manage ad space reservation*” and “*Manage ad order*” are modelled on the basis of six reaction rules. Variables in the parameter list of a predicate are prefixed with a question mark. The activities “*Confirm ad order*” and “*Update ad size*” require an action by a human internal agent Publication Secretary and are therefore *human activities*, while all the other activities are to be performed by automated software agents and are thus *automated activities*.

- R1:** Upon receiving from the MediaAgency a request to reserve ad space, the activity “*Manage ad publishing*” is started.
- R2:** Upon the start of the subactivity “*Check ad space*”, if the Publication has enough ad space on the date requested – determined by evaluating the intensional predicate `Issue.hasSpaceFor(?RsrvRequest)` – the Publication sends a confirmation to the Media Agency. Otherwise, if the Publication does not have sufficiently ad space on the date requested, the Publication sends a rejection to the Media Agency.
- R3:** Upon the start of the subactivity “*Reserve ad space and commit*”, the Publication creates the corresponding ad space reservation (i.e. an instance of `AdOrder` with the status `isReserved`), and commits towards the Media Agency to publish the ad.
- R4:** Upon receiving an ad order from the Media Agency, the subactivity “*Manage ad order*” is started.
- R5:** Upon the approval of the ad order by the Publication Secretary, the Publication changes the status of the corresponding ad space reservation to `isOrdered`, and sends a confirmation to the Media Agency.
- R6:** Upon the insertion of the actual ad size of the ad published by the Publication Secretary, the corresponding object instance of the type `AdSize` is updated.

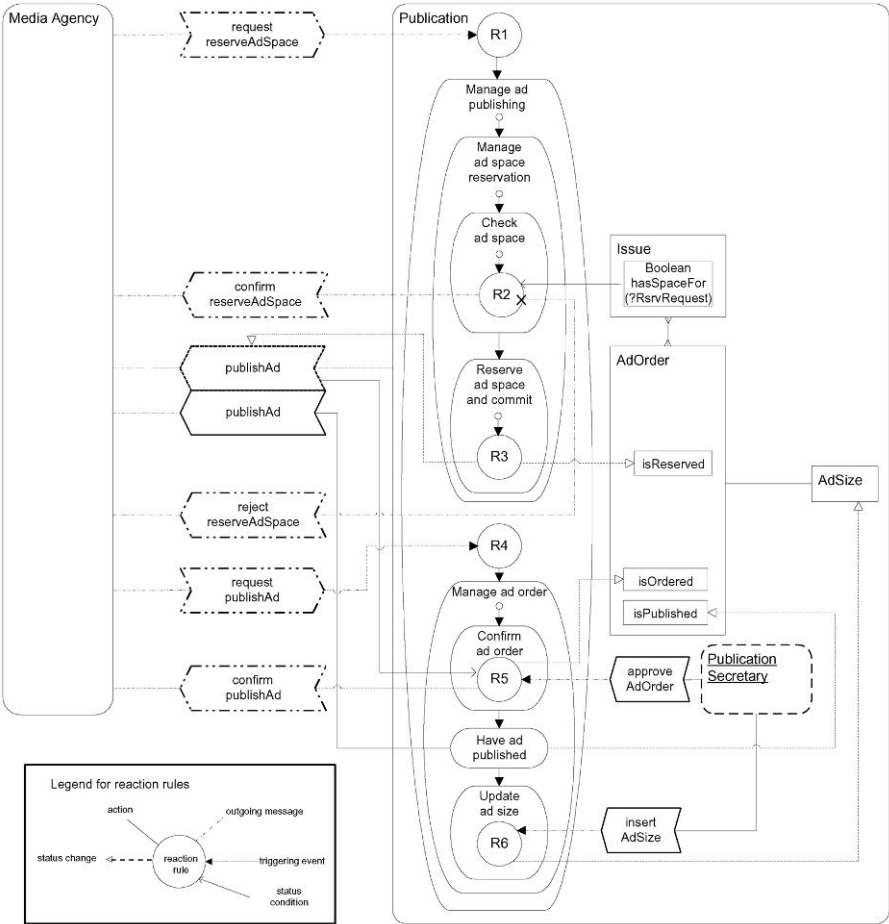


Fig. 5. Visualization of activities and reaction rules by an activity diagram.

3 Related Work

In the paper [22] a general methodology for agent-oriented analysis and design is presented. The methodology proposed deals with both the macro-level (societal) and the micro-level (agent) aspects of systems. In the analysis phase of the methodology, the *roles* in the system are identified and the patterns of interaction that occur in the system between various roles are recognized. The functionality of each role is defined by its liveness and safety responsibilities. *Liveness responsibilities* are those that say “something will be done”, e.g. “whenever the coffee machine is empty, fill it up”. *Safety responsibilities* relate to the absence of some undesirable condition arising, e.g. “the coffee stock should never be empty”. In the design phase, the liveness and safety

responsibilities are respectively mapped to agents' *services* and *pre-* and *postconditions* on each service. The difference from our work is that the methodology proposed in [22] is a software engineering approach, while our approach is aimed at creating business information systems and therefore contains notions needed for it like intentional dependencies, goals, actions/events, and commitments/claims. The Tropos methodology described in [23] uses the models of *i** all along the lifecycle of an information system as a foundation to model both early and late requirements, architectural and detailed design, and implementation. An advantage of our approach compared to Tropos is that our approach moves with the actors and tasks identified at the stage of analysis to the stage of design where it distinguishes between human and automated activities, while in Tropos the activities modelled at design by UML [28] and AUML [27] are to be performed by software agents, and human activities disappear somewhere. Also, the target implementation model of Tropos is the BDI agent architecture [11, 14] which may be too restrictive and inefficient for implementing real-life information systems.

4 Conclusions and Future Work

The main contribution of our work is offering a methodology for business modelling and designing global information systems that bridges the gap between the perspectives of descriptive and prescriptive modelling by means of (meta)commitments. We propose to start engineering of information systems in support of inter-enterprise business processes from descriptive modelling of intentional dependencies between actors of the problem domain by SD models of *i**, and to continue by analyzing means-ends reasoning of individual actors using SR models of *i**. After that, the intentional dependencies are represented as metacommitments that serve as a basis for modelling lifecycles of first-order commitments by interaction frames of AOR modelling. We also suggest to refine the tasks identified by SR models by representing them as AOR activity diagrams which enables to identify the atomic actions together with their triggering events, and pre- and postconditions, and to transform all four to reaction rules of AOR modelling. Since reaction rule is a generic notion (e.g. triggers in relational databases [9] are reaction rules of a kind), reaction rules can be rather straightforwardly mapped to specific agent architectures and/or platforms like BDI [11, 14], vivid agents [10], AgentBuilder® [1], or the behaviour-based agent model of the JADE platform [3], compliant with the FIPA ACL [18]. Our approach thus enables to choose between different implementation architectures and platforms.

The notion of commitment also enables to capture various *business rules* [5, 6, 13] in a more natural way. This aspect of our work is addressed in [7] and [15]. We plan to add models for exception handling where exceptions are captured as violations of commitments. We also intend to find a suitable formalism for representing possible conflicts and inconsistencies in the agent's commitments, and for checking the agent's commitments on them in the precondition parts of reaction rules. Our future aims also include further *verification* and *validation* of our work.

References

1. AgentBuilder®, <http://www.agentbuilder.com>
2. Wagner, G.: Agent-Oriented Enterprise and Business Process Modeling. In: Proceedings of the First International Workshop on Enterprise Management and Resource Planning Systems (EMRPS'99), Venice, November 1999.
3. Bellifemine, F., Poggi, A., Rimassa, G.: Developing multi-agent systems with FIPA-compliant agent framework. *Software – Practice and Experience* **31** (2001) 103-128.
4. Wagner, G.: Foundations of Knowledge Systems with Applications to Databases and Agents. Kluwer Academic Publishers, Boston Dordrecht London (1998).
5. Ross, R. G.: The Business Rule Book: Classifying, Defining and Modeling Rules. Second Edition. Boston, Massachusetts, Database Research Group, Inc. (1997).
6. Defining Business Rules - What Are They Really? The Business Rules Group, formerly known as the GUIDE Business Rules Project, Final Report. Revision 1.3, July 2000. Prepared by D. Hay and K. A. Healy. Available at http://businessrulesgroup.org/first_paper/br01c0.htm
7. Taveter, K., Wagner, G.: Agent-Oriented Enterprise Modeling Based on Business Rules. In: Proceedings of the 20th International Conference on Conceptual Modeling (ER'2001), Yokohama, Japan, November 27-30, 2001. Springer-Verlag, Berlin Heidelberg New York (2001), forthcoming.
8. Wagner, G.: Agent-Oriented Analysis and Design of Organizational Information Systems. In: Barzdins, J., Caplinskas, A. (eds.): Databases and Information Systems. Proceedings of the Fourth IEEE International Baltic Workshop on Databases and Information Systems, 1-5 May 2000, Vilnius, Lithuania. Kluwer Academic Publishers, Boston Dordrecht London (2000).
9. Widom, J., Ceri, S. (eds.): Active Database Systems: Triggers and Rules for Advanced Database Processing. Morgan Kaufmann Publishers, Inc., San Francisco (1996).
10. Wagner, G.: Vivid Agents - How they Deliberate, How They React, How They Are Verified. Extended version of: Wagner, G.: A Logical and Operational Model of Scalable Knowledge- and Perception-Based Agents. In: Van de Velde, W., Perram, J. W. (eds.): Agents Breaking Away. Proceedings of MAAMAW'96. Lecture Notes in Artificial Intelligence, Vol. 1038. Springer-Verlag, Berlin Heidelberg New York (1996). Available At <http://tmitwww.tm.tue.nl/staff/gwagner>
11. Georgeff, M. P., Lansky, A.: Reactive reasoning and planning. In: Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), Seattle, Washington, USA (1987) 677-682.
12. Antikainen, H., Bäck, A.: Challenges of electronic advertising processes in newspapers. Technical report. VTT Information Technology, Espoo, Finland (2000).
13. Herbst, H.: Business Rule-Oriented Conceptual Modeling (Contributions to Management Science). Springer-Verlag Berlin Heidelberg New York (1997).
14. Huber, M. J.: Jam Agents in a Nutshell, <http://members.home.net:80/marcush/IRS/>
15. Taveter, K., Wagner, G.: Combining AOR Diagrams and Ross Business Rules' Diagrams for Enterprise Modeling. In: Proceedings of the Second International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2000), 5-6 June 2000, Stockholm (Sweden) and 30 July 2000, Austin (Texas, USA). iCue Publishing, Berlin (2000).
16. Hammer, M., Champy, J.: Reengineering the Corporation. New York, Harper Collins (1993).
17. Yolum, P., Singh, M.P.: Synthesizing Finite State Machines for Communication Protocols. North Carolina State University Technical Report TR-2001-06 (2001).
18. Foundation for Intelligent Physical Agents (FIPA), <http://www.fipa.org>
19. Knowledge Query and Manipulation Language (KQML), <http://www.cs.umbc.edu/kqml/>

20. Davenport, T. H.: *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press (1992).
21. Yu, E.: *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, Department of Computer Science, University of Toronto (1994).
22. Wooldridge, M., Jennings, N., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems* **3** (2000) 285-312.
23. Castro, J., Kolp, M., Mylopoulos, J.: A Requirements-Driven Development Methodology. In: *Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01)*, Interlaken, Switzerland, June 4-8, 2001. Springer-Verlag Berlin Heidelberg New York (2001).
24. Venkatraman, M., Singh, M. P.: Verifying Compliance with Commitment Protocols: Enabling Open Web-Based Multiagent Systems. *Autonomous Agents and Multi-Agent Systems* **3** (2000) 217-236.
25. Yu, E. S. K., Mylopoulos, J.: From E-R to 'A-R' - Modelling Strategic Actor Relationships for Business Process Reengineering. *International Journal of Intelligent and Cooperative Information Systems* **2/3** (1995) 125-144.
26. Singh, M.: An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts. *Artificial Intelligence and Law* **7** (1999) 97-113.
27. Odell, J., Van Dyke Parunak, H. Bernhard, B.: Representing Agent Interaction Protocols in UML. In: *Proceedings of the First International Workshop on Agent-Oriented Software Engineering (AOSE-2000)*, Limerick, Ireland, June 2000.
28. Unified Modelling Language (UML), <http://www.omg.org/uml/> 29. Ould, M. A.: *Business processes: modelling and analysis for re-engineering and improvement*. John Wiley & Sons, Chichester (1995).

Process Patterns to Generate E-commerce Systems

Prasad Jayaweera, Paul Johannesson, and Petia Wohed

Department of Computer and Systems Sciences
Stockholm University and Royal Institute of Technology
Electrum 230, SE-164 40 Kista, Sweden
{prasad, pajo, petia}@dsv.su.se

Abstract. In electronic commerce, two fundamental types of models are business models and process models. While a business model is concerned with value exchanges between actors, a process model describes the procedural realization of business requirements. There is a need for methodological guidelines and tool support to move from a business model to a process model, which enables design decisions to be based on requirements captured in the business model. This paper addresses a systematic transformation of business models to process models. We propose a designer assistant that systematically aids a designer in generating a process model in an executable process modeling language.

1 Introduction

When building e-commerce systems, two types of models are fundamental: business models and process models, [4]. The purpose of a business model is to describe the fundamental business aspects of the e-commerce system to be built. Thus, a business model describes which actors are involved, what the actors offer each other, and what activities they perform when producing and consuming offerings. The central concept in a business model is that of *value*, and the model describes how value is exchanged between actors [9]. This can be contrasted to a process model, which aims at describing the operational and procedural aspects of a process and specifies the control flow of the activities carried out in a process. A process model specifies the actors involved in the operations, which activities they perform as well as the sequencing of these activities. Thus, a business model defines the *what* in an e-commerce system, while a process model defines the *how*.

A business model can be seen as more basic than a process model as it specifies the declarative aspects of an e-commerce system. A natural question is, therefore, whether it is possible to move from a business model to a process model in a systematic way. Methodological support for this task would provide several benefits: support for identifying design alternatives in process modeling, support for motivating design choices, and a clarification of the relationships between the declarative business model and the procedural process model.

In this paper, we argue that it is possible to systematically move from a business model to a process model, and we suggest methodological guidelines and modeling techniques that can assist a designer in the task. A starting point of the method

proposed is that much of the procedural aspects of a process model concern communication among actors. This communication is carried out in order to establish commitments among the actors to perform exchanges of values. The commitments are created by speech acts, and the control flow in a process is determined by the interleaving of these speech acts with each other and with the value exchanging activities.

The proposed guidelines and techniques are based on four building blocks:

1. Business Model
2. BML, a formal and executable process definition language
3. Predefined Process Patterns
4. Automated Designer Assistant

The paper is an extension of process patterns in our work [6] and is structured as follows. Section 2 outlines relevant research in the area and introduces our conceptual framework. Section 3 introduces the identified process patterns built on communicative acts. Section 4 describes the automated designer assistant that supports the task of creating process models. Section 5 concludes the paper and suggests directions for further research.

2 Related Research and Conceptual Framework

The approach proposed in this paper is a combination of elements from the Language Action approaches to information systems design, which focus on communication aspects, and elements from the work by Gordijn et. al. [4], who focus on the transfer of values between actors and explicitly distinguish between business and process models.

Language Action approaches to information systems design build on Austin's Speech Act Theory [1], which acknowledges speech acts as a special action category. A speech act is defined as an action (changing the universe of discourse) performed by a speaker and grasped by a recipient. Searle [12] further develops the theory by introducing a taxonomy of five different kinds of speech acts, namely: *assertive*, *directive*, *commissive*, *expressive*, and *declarative*. Well-known Language Action approaches are Action Workflow Loop [10], Business Action Theory (BAT) [3], and Dynamic Essential Modeling of Organization (DEMO) [2]. We have adopted Action Workflow Loop (Fig. 1) for the definition of proposed process patterns.

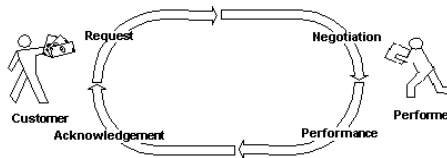


Fig. 1. Action Workflow Loop with four phases

The conceptual framework we use is based on UMM's economic model describing resources, events and agents (REA model) [15], [5]. In order to make the model suitable also for communication aspects, we have extended it by concepts from

speech act theory. Furthermore, we include a number of notions proposed by Weigand et. al. [16] used for distinguishing between different levels of communication.

The extended REA model (Fig. 2) describes business events and business entities as well as their structure and relationships. The basic business entities are Agreement, Economic Resource Type and Economic Resource. An Agreement is an arrangement between two Partner Types. Furthermore, an Agreement with any economic commitment is called Economic Contract and specifies the economic exchanges to occur, which is represented by the entity Commitment. An example of an Economic Contract is a purchase order where each order line is a Commitment. A Commitment is an obligation from one Partner Type to another to transfer the specified Economic Resource Type, and it is fulfilled through an Economic Event.

The most central concepts in our approach [the shaded rectangles in (Fig. 2)] are the following:

Partner: A partner is an independent economic and/or legal entity, e.g., John Doe, Stockholm University.

Economic resource: An economic resource is a quantity of something of value that is under the control of a partner, e.g., a car.

Economic event: An economic event is the transfer of the control of an economic resource from one partner to another partner, e.g., the ownership change of a car.

Duality: The corresponding term used by Gordijn, [4], is “value offering” and is used to represent the relationship between two economic events, T1 and T2, that satisfy the condition: if T1 is an economic event transferring an economic resource from partner A1 to partner A2, then T2 is the corresponding economic event transferring an economic resource from A2 to A1. The intuition is that the Duality represents the reciprocity between the economic events - one partner providing another partner with something of value and receiving something of value in return. An example of Duality is a car purchase where a car ownership is transferred from a retailer to a customer and the corresponding payment is provided.

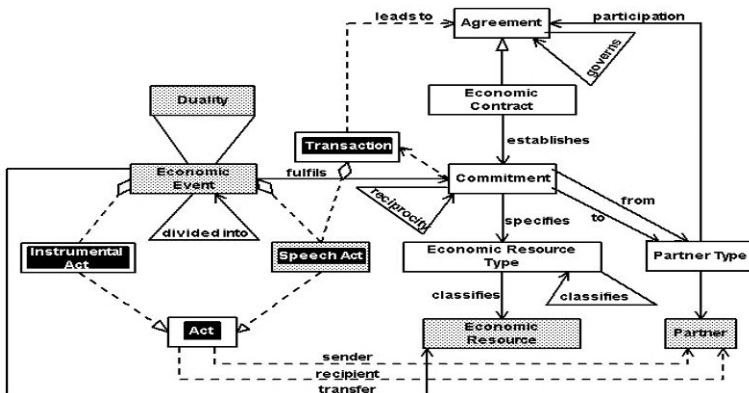


Fig. 2. An extended UMM economic model over REA (The black labels, as well as the dotted relationships, mark the extensions of the original model. The shaded rectangles represent the central concepts in our approach)

We also allow an economic event to be **divided into** several Economic Events. This may occur when a monetary transfer is done, e.g., a payment is divided into a down payment and a final payment, or when a delivery of a large order is divided into

several shipments. Furthermore, applying speech act theory, we claim that an economic event consists of one or more **Speech Act(s)** and, optionally, an Instrumental Act. An Instrumental Act represents the physical transfer of an Economic Resource from one Partner to another one, whereas Speech Act models the communication between the partners. Finally, applying Weigand's levels for communication patterns [16], some pairs of speech acts build Transactions, e.g., a request/commit. Some Transactions lead to Agreements, while others directly result in Commitments.

Now, we define a **Business Model** as a triple $\langle PT, ER, DT \rangle$, where

- PT - a set of partner types
- ER - a set of economic resource types
- DT - a set of duality types

An example of a business model is shown in (Fig. 3). This model represents four partner types: eCaterer, Customer, WineSupplier, and FoodSupplier. The customer orders meals from the eCaterer who purchases wine from the WineSupplier and food from the FoodSupplier and prepares the meal. In this example, $PT = \{ \text{Customer, eCaterer, WineSupplier, FoodSupplier} \}$, $ER = \{ \text{Meal, Money, Wine, Food} \}$, $DT = \{ \langle \text{eCaterer, Customer, Meal} \rangle, \langle \text{Customer, eCaterer, Money} \rangle, \langle \text{eCaterer, WineSupplier, Money} \rangle, \langle \text{eCaterer, FoodSupplier, Money} \rangle, \langle \text{WineSupplier, eCaterer, Wine} \rangle, \langle \text{FoodSupplier, eCaterer, Food} \rangle \}$. DT represents the dualities: MealSupply, FoodPurchase, and WinePurchase.

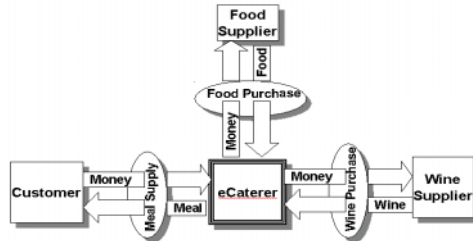


Fig. 3. Business Model for eCatering Example

The process definition language BML (has similarities to SDL [11]) is the formal and executable language based on communicating state machines used in our work. Interested readers are referred to [7]. BML diagrams are designed from one partner's (called here, the base partner) perspective and model messages sent from and received by the base partner.

In our work we have used BML symbols: *convex and concave hexagon* for send and receive message respectively, *diamond* for business decision, *ellipses* for start, wait and stop states, and finally *stick man* for a partner involved (Fig. 5), (Fig. 6). We have extended, the original BML syntax by allowing wait states to receive messages and also to send message out from *Stop* state, simply for the compactness and clarity of process model but preserving a straight forward mapping between the original syntax and our extended version.

3 Generic Process Patterns

In this section, we introduce and describe four patterns for e-commerce in the form of BML diagrams. We hypothesize that most process models for e-commerce can be expressed as a combination of these four diagrams.

- *Economic Resource Requesting Diagram (ERRD)*. This diagram is an action-workflow loop from the perspective of the requesting actor, i.e. when the requesting actor is the base partner. For example, a base partner's purchase order enacts one ERRD instance to acquire supplier delivery.
- *Economic Resource Offering Diagram (EROD)*. This diagram is an action-workflow loop from the perspective of the supplying actor, i.e. when the supplying actor is the base partner. For example, a customer's order enacts one EROD instance for the fulfillment of customer delivery.
- *Providability Checking Diagram (PCD)*. This diagram models the reservation and booking of resources needed for carrying out an economic event.
- *Counter Offer Diagram (COD)*. This diagram handles communication carried out in order to identify alternatives to a request that could not be fulfilled.

These diagrams and their intercommunication are shown in (Fig. 4). A typical scenario would be the following: a customer orders an economic resource by initiating an EROD instance. The EROD checks the availability of required resources by starting a PCD instance, which communicates with a supplier first directly and then through an ERRD instance. PCD first reserves and then books the necessary economic resources. Reservation, holding resources for a very short time, is weaker compared to booking, which leads to ordering [7]. If the request by the customer cannot be fulfilled, counter offer management can be initiated through a COD instance.

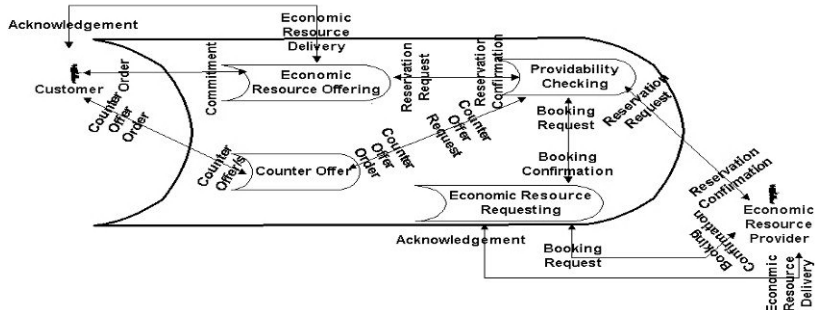


Fig. 4. BML Static Diagram for Generic four Process Patterns

3.1 Economic Resource Requesting Diagram (ERRD)

An Economic Resource Requesting Diagram (Fig. 5) models a situation where the base partner receives an economic resource from another actor, called supplier.

Step 1: The base partner sends an order to the supplier requesting an economic resource.

Step 2: The supplier sends a reply for the order made by the base partner.

Step 3: The supplier's reply is interpreted either as a rejection or as a commitment to fulfil the order.

Step 4: The supplier declares the delivery of the ordered economic resource.

Step 5: The base partner acknowledges the receipt of the requested economic resource.

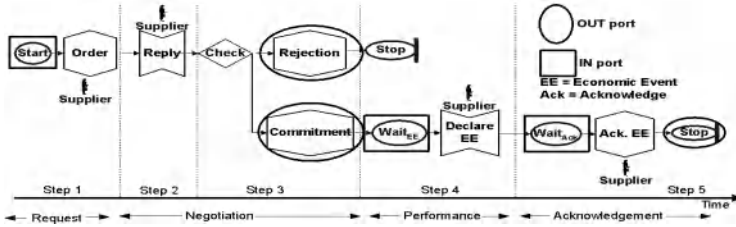


Fig. 5. Economic Resource Requesting Diagram (ERRD)

In this basic BML diagram, there are some explicit positions where inter diagram communication is possible. Sending out positions are called *OUT ports* while receiving positions are called *IN ports*.

3.2 Economic Resource Offering Diagram (EROD)

An Economic Resource Offering Diagram models a situation where the base partner supplies another actor, called customer, with an economic resource (Fig. 6). In this case, we follow the suggestion by James Taylor in [13] and introduce additional qualification steps, where the base partner acquires *direct* and *indirect* means required for carrying out the requested action. Direct means are resources from suppliers needed to carry out the requested action. Indirect means are resources from the customer corresponding to the duality of the economic event.

Step 1: The customer sends an order for an economic resource to the base partner.

Step 2: The qualification steps are handled by choosing one of three alternatives. In the first one, there is no order between acquiring direct and indirect means. In the second one, direct means are acquired before indirect means. In the third one, indirect means are acquired before direct means. This includes one or more synchronize request-reply to acquire all necessary means prior to commit recipient with a delivery.

Step 3: The replies received when acquiring direct means are evaluated, and the base partner either rejects the customer's order or commits to deliver the customer order.

Step 4: The base partner declares the delivery of the ordered economic resource to the customer.

Step 5: The customer acknowledges the receipt of the ordered economic resource.

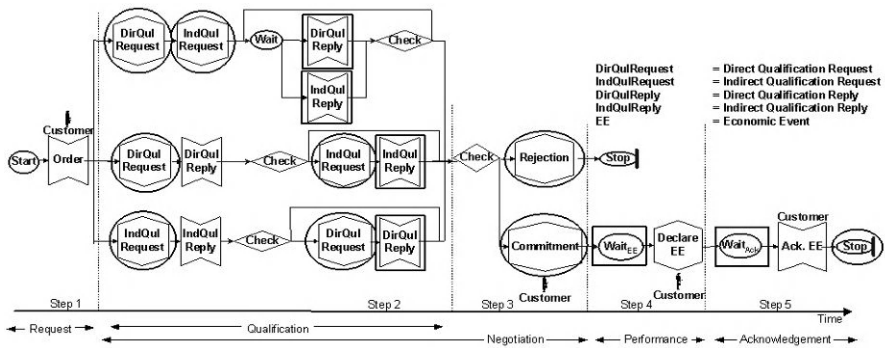


Fig. 6. Economic Resource Offering Diagram (EROD)

All the steps prior to sending the message “Commitment” of ERRD and EROD can be viewed as a commitment establishment process between the involved partners, while the succeeding steps can be viewed as a commitment fulfillment process.

Note that there is an asymmetry between the economic resource requesting and offering diagram. The reason for this is that the qualification steps are relevant only when the base partner has to supply an economic resource. The additional qualification steps may lead to two other fundamental process patterns Providability Checking and Counter Offer that are described below. In the case of an economic resource requested by the base partner, these two process patterns are not under her control.

3.3 Providability Checking Diagram (PCD)

The Providability Checking Diagram (Fig. 7) models a situation where direct means for an economic resource offering is first reserved and then booked. In the preliminary reservation, PCD communicates directly with a supplier. When the booking request is received from EROD, it invokes an ERRD to make the formal ordering. The Providability Checking Diagram (PCD) may invoke counter offer handling when the base partner is not capable of providing the original order.

- Step 1:** A reservation request is received from the Economic Resource Offering Diagram (EROD).
- Step 2:** The reservation request is sent to a supplier and a reservation reply is received.
- Step 3:** The reservation reply is evaluated either to a counter offer exception, which is to be handled by a Counter Offer Diagram (COD) for the economic resource, or to a reservation confirmation that is to be sent to an EROD.
- Step 4:** A booking request from EROD is sent to an ERRD.
- Step 5:** When the booking confirmation from the ERRD is received, EROD is acknowledged with the received booking confirmation.

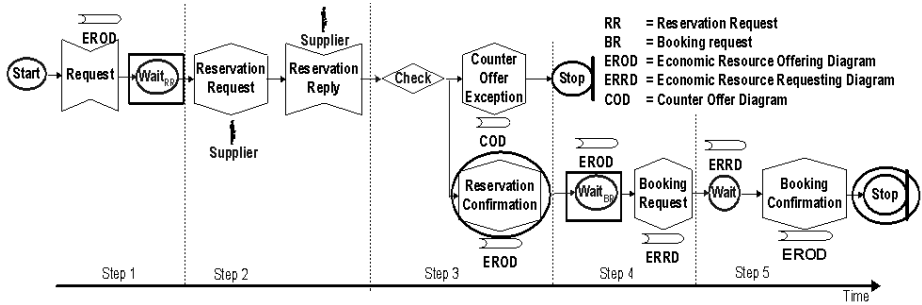


Fig. 7. Providability Checking Diagram with marked IN and OUT ports (PCD)

3.4 Counter Offer Diagram (COD)

A Counter Offer Diagram (Fig. 8) models the management of counter offers, which are received from a PCD. A COD communicates with a Counter Offer Provider and with a customer in order to identify an alternative for the original request made by the customer.

- Step 1:** A counter offer request is received from PCD.
- Step 2:** A counter offer request is sent to the Counter Offer Provider (COP) and the reply is received. (A COP is a process that creates counter offers; it may use advanced algorithms and/or include human involvement.)
- Step 3:** The response received from COP is evaluated either to a rejection that is to be notified to the customer or to a set of counter offers that the customer may choose among.
- Step 4:** The customer can reply with a rejection, request a selected offer, or request an alternative offer.
- Step 5:** If the customer rejects the counter offer, the rejection is directed to a relevant handler process here called Rejection Recorder. If the customer requests a selected offer, then the request is sent to new PCD instance. If the customer requests an alternative, the request is sent to a new COD instance.

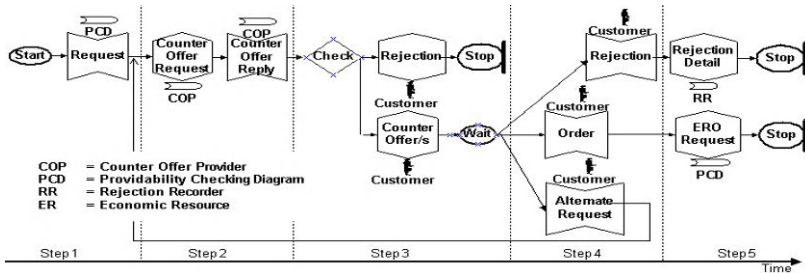


Fig. 8. Counter Offer Diagram with explicit interactions (COD)

These four basic process patterns can be group into direct process patterns and intermediate process patterns. The first two patterns (ERRD and EROD) fall into

direct process patterns where initial interaction is with external partner and also notice that there are no inter-diagram communications identified at the definition level. The last two patterns (PCD and COD) fall into intermediate process patterns where enactment is caused by one instance of the direct process pattern. Therefore at the definition level, some inter-diagram communications have marked of intermediate process patterns.

3.5. Process Model

A process can be modeled by a set of Economic Resource Requesting Diagrams, Economic Resource Offering Diagrams, optionally Providability Checking and Counter Offer Diagrams – such a set is called a *process model*. The basic structure of the diagrams in a process model can be derived simply from a business model. However, the communication among the diagrams is not uniquely determined by the business model, but may vary depending on the requirements for the process. How to determine this communication, i.e. how to move from a business model to a process model is the main topic of the next section.

4 A Designer Assistant

In this section, we will show how a business model can be transformed and extended into a process model in a systematic way. Moving from a business model to a process model is not a trivial task but requires a large number of design decisions. In order to support a designer in this task, we propose an automated designer assistant that guides the designer through the task by means of a sequence of questions (a similar designer assistant has been proposed for the area of conceptual modeling by Wohed [17]). The questions can be divided into four phases, (Fig. 9).

Phase 1. The designer builds the business model, identifies the base partner, i.e. the organisation from whose perspective the system is to be built, and the customer of the process.

Phase 2. The designer establishes a (partial) order between the economic events of the business model.

Phase 3. The designer introduces the communicative acts needed to handle the economic events and establishes a (partial) order between them.

Phase 4. From the output of phase 3, a process model is automatically derived.

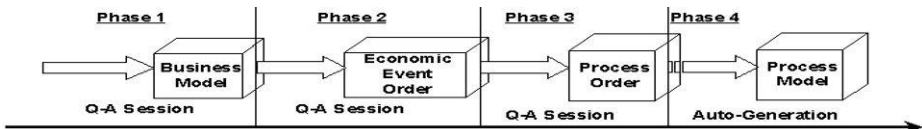


Fig. 9. Four phased approach to process models

4.1 Phase 1 - Business Model

In the first phase, the designer builds a business model and specifies the organization for which the e-commerce system is to be developed, as well as its customer. Due to the space limitations, this phase is omitted and interested readers are referred to [6]. However, in order to clarify, we exemplify the result from this phase by referring back to (Fig. 3) where the business model for the described example is shown. We will further use this example as a running example through the rest of the paper.

4.2 Phase 2 - Economic Event Order (EEO)

In this phase, the designer starts to construct an order between the activities of the process. First, the designer takes into account only economic events while disregarding the communicative acts that co-ordinate the process. By considering only the order of the economic events in this phase, the designer can concentrate on the main business logic and postpone until later more detailed design decisions about the coordination of communicative acts.

The designer first has to decide whether an economic event must or can be divided into parts; such a part is called an *atomic economic event*. The first question is therefore:

1. *What are the atomic economic events of each economic event?*

After having identified and named the atomic economic events, the designer is prompted to order them by determining the dependencies that exist between them. In an e-commerce context, we identify two main types of dependencies: trust dependencies and flow dependencies.

A *trust dependency* occurs between two atomic economic events within the same duality, e.g. that a product must be paid before it can be delivered. A trust dependency expresses the level of trust between the actors involved in a duality, e.g. requesting a down payment expresses low trust. A *flow dependency*, [8], occurs between two atomic economic events in different dualities and expresses that the economic resource obtained by one of the economic events is needed for the other economic event. A simple example is that a retailer has to obtain a product from an importer before delivering it to a customer. In order to identify trust and flow dependencies, the following question is posed.

2. *How do you order the atomic economic events?*

An example of answers to the questions 1 and 2 is given in (Fig. 10).

4.2.1 Reservation and Booking

When the offered economic resource is assembled from more than one economic resources, the acquisition of direct means for each can be ordered by getting answers for the questions below. The economic event order can be completed with reservation and booking as follows. An example answer to the questions is given in (Fig. 11).

1. What are the Atomic Economic Events (AEEs) of each Economic Event (EE)?

Economic Event	Atomic Economic Event	Abbreviation
<eCaterer, Customer, Meal>		MealToCustomer (MTC)
<Customer, eCaterer, Money>	<Customer, eCaterer, DownPay>	DownPayFromCustomer (DPFC)
	<Customer, eCaterer, FinalPay>	FinalPayFromCustomer (FPFC)
<WineSupplier, eCaterer, Wine>	<eCaterer, WineSupplier, DownPay>	WineToCateringCompany (WTCC)
	<eCaterer, WineSupplier, FinalPay>	DownPayToWineSupplier (DPTWS)
<FoodSupplier, eCaterer, Food>	<eCaterer, WineSupplier, FinalPay>	FinalPayToWineSupplier (FPTWS)
		FoodToCateringCompany (FTCC)
<eCaterer, FoodSupplier, Money>		FinalPayToFoodSupplier (FPTFS)

2. How do you order the Economic Events and the Atomic Economic Events?

(Add only < symbol in the matrix)

	MTC	DPFC	FPFC	WTCC	DPTWS	FPTWS	FTCC	FPTFS
MealToCustomer (MTC)								
DownPayFromCustomer (DPFC)	<		<					
FinalPayFromCustomer (FPFC)								
WineToCateringCompany (WTCC)	<							
DownPayToWineSupplier (DPTWS)				<				
FinalPayToWineSupplier (FPTWS)					<			
FoodToCateringCompany (FTCC)	<							
FinalPayToFoodSupplier (FPTFS)								<

The table shall be read in the following way: row header, cell, column header, e.g. the first cell on the seventh row gives FoodToCateringCompany precedes MealToCustomer.

Fig. 10. Questions and answers for the example in phase 2

3a. Do you reserve economic resource ER_1 before reserving economic resource, ER_2 ?

3b. Do you book economic resource ER_1 before booking economic resource, ER_2 ?

3c. Do you book economic resource ER_1 before reserving economic resource ER_2 ?

The resulting partial order (EEO) from phase 2 is the following (“<” means precedes):

EEO = {MealToCustomer < FinalPayFromCustomer, DownPayFromCustomer < MealToCustomer, DownPayFromCustomer < FinalPayFromCustomer, WineToCateringCompany < MealToCustomer, WineToCateringCompany < FinalPayToWineSupplier, DownPayToWineSupplier < WineToCateringCompany, DownPayToWineSupplier < FinalPayToWineSupplier, FoodToCateringCompany < MealToCustomer, FoodToCateringCompany < FinalPayToFoodSupplier, BookWine < BookFood, ReserveWine < ReserveFood}

Such a partial order between atomic economic events is called an *economic event order*. It expresses the order between the most important activities in the process and disregards coordinative activities.

3. For each Atomic Economic Event, identified the ones needed with counter offer handling.

- Wine Supply with counter offer handling
- Food Supply with counter offer handling

Column	Wine	Food
Row		
Wine		3a. BookWine < BookFood? Yes 3b. ReserveWine < ReserveFood? Yes 3c. BookWine < ReserveFood? No
Food	3a. BookFood < BookWine? -	
	3b. ReserveFood < ReserveWine? -	
	3c. BookFood < ReserveWine? -	

i.e. Get answers for three questions;

Book(row) < Book(column)?

Reserve(row) < Reserve(column)?

Book(row) < Reserve(column)?

considering a pair at a time.

Fig. 11. Questions and Answers for Reservation and Booking

4.3 Phase 3 - Process Order

In phase 3, the designer will extend the economic event order from phase 2 by specifying dependencies between communicative acts. A starting point for this task is that for each atomic economic event, there will be one action-workflow loop. The designer has to determine the interactions between the loops given by all the atomic economic events. Like earlier, the designer assistant supports this task through a number of questions. The intuition behind several of these questions is, roughly expressed, the following: Before an actor does something of value to another actor, it will check whether that actor has deserved it. By doing “something of value to another actor” is meant to carry out an economic event, to commit to carry out an economic event, or to initiate the acquisition of means needed to carry out an economic event. The expression “check whether that actor has deserved it” has to do with the fact that an economic event from an actor A to an actor B always is accompanied by another economic event from B to A; recall that these two economic events together constitute one duality. The expression states that before actor A is prepared to carry out its economic event (or some preparation to it) to B, it will check whether B has done its corresponding economic event (or some preparation). Note that this check will be done only if the economic event order so prescribes. Furthermore, there are questions for ensuring that all required means for carrying out an economic event have been obtained.

In order to formulate the questions, we need to distinguish between an *incoming atomic economic event* (AEE), where the base partner receives an economic resource, and an *outgoing AEE*, where the base partner supplies an economic resource.

If In is an incoming AEE and Out an outgoing AEE within the same duality, and $In < Out$ in the economic event order, ask:

- 4a. Do you require that In be performed before you commit to perform Out?
- 4b. Do you require a commitment for In before you commit to perform Out?

Furthermore, if In is an incoming AEE and Out an outgoing AEE within the same duality, and Means is an incoming AEE in another duality, and $In < Out$, and $Means < Out$ in the economic event order, ask:

- 5a. Do you require that Means be performed before you commit to perform Out?
- 5b. Do you require a commitment for Means before you commit to perform Out?
- 5c. Do you require that In be performed before you request Means?

An example of answers to these questions is given in (Fig. 12). All answers will result in an extension to the economic event order from phase 2, which also includes ordering between communicative acts. Such an order is called a *process order*. In this case, we arrive at a process order PO:

```
PO = EEO U
{decl(DownPayFromCustomer) < com(MealToCustomer), com(WineToCateringCompany) < com(FinalPayToWineSupplier),
com(FoodToCateringCompany) < com(FinalPayToFoodSupplier), com(WineToCateringCompany) < com(MealToCustomer),
decl(DownPayFromCustomer) < dir(WineToCateringCompany), com(FoodToCateringCompany) < com(MealToCustomer),
decl(DownPayFromCustomer) < dir(FoodToCateringCompany)}
```

EEO is the economic event order derived in phase 2 for the example. Abbreviations for communicative acts used in PO are: ‘*dir*’ for directive, ‘*com*’ for commissive, and ‘*decl*’ for declarative.

- 4a. Do you require *DownPayFromCustomer* to be performed before you commit to perform *MealToCustomer*? (Build) No
It is unnecessary to check it as the answer to 3 already implies it.
- 4a. Do you require *WineToCateringCompany* to be performed before you commit to perform *FoodToMealSupplier*? (Build) No
- 4b. Do you require a commitment for *WineToCateringCompany* before you commit to perform *FoodToMealSupplier*? (Build) No
- 4a. Do you require *FoodToCateringCompany* to be performed before you commit to perform *FoodPayToMealSupplier*? (Build) No
- 4b. Do you require a commitment for *FoodToCateringCompany* before you commit to perform *FoodPayToMealSupplier*? (Build) No
- 5a. Do you require that *WineToCateringCompany* be performed before you commit to perform *MealToCustomer*? (Build) No
- 5b. Do you require a commitment for *WineToCateringCompany* before you commit to perform *MealToCustomer*? (Build) No
- 5c. Do you require *DownPaymentFromCustomer* to be performed before you request *WineToCateringCompany*? (Build) No
- 5a. Do you require that *FoodToCateringCompany* be performed before you commit to perform *MealToCustomer*? (Build) No
- 5b. Do you require a commitment for *FoodToCateringCompany* before you commit to perform *MealToCustomer*? (Build) No
- 5c. Do you require *DownPaymentFromCustomer* to be performed before you request *FoodToCateringCompany*? (Build) No

Fig. 12. Example of answers to question in section 4.3

4.4 Phase 4 - Mapping Process Order to BML Process Model

The basis in generation of a process model is that for each atomic economic event, there is an action workflow loop represented in a process model by one of the direct process patterns, i.e. an ERRD or an EROD. The process order, identified through the second and the third step is necessary for identifying the inter-diagram connections as well as whether it is necessary to include an intermediate process pattern, i.e., a PCD or a COD or not. Basically, each inequality identifies one inter-diagram connection. The formal rules for this mapping are, due to space limitation, omitted here, but may be found in [14].

However, to give a basic intuition of the logic behind them, we are exemplifying the effect of a couple of the mapping rules. For example an inequality telling that the wine has to be delivered to the catering company before the catering company can in turn deliver the meal to the customer ($\text{WineToCateringCompany} < \text{MealToCustomer}$), should result in a connection from the stop state of the ERRD modeling the caterer's wine request process, to the wait for confirmation state in the EROD modeling the caterer's meal offering process (Fig. 13). This inter-diagram connection indicates that the wine delivery to catering company process has to be completed before the meal delivery process can go on, which indeed is the semantic of the inequality.

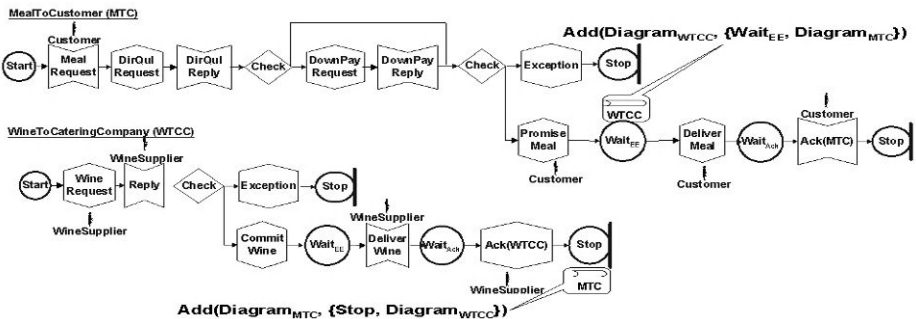


Fig. 13. BML process segment (MealToCustomer and WineToCateringCompany processes)

Another example of an inequality telling that getting down payment from customer before ordering wine from wine supplier [$\text{decl}(\text{DownPayFromCustomer}) < \text{dir}(\text{WineToCateringCompany})$], will connect stop state of ERRD modeling down

payment from customer and start state of ERRD modeling wine to catering company, indicating that completion of former enacts the latter process instance (Fig. 14).

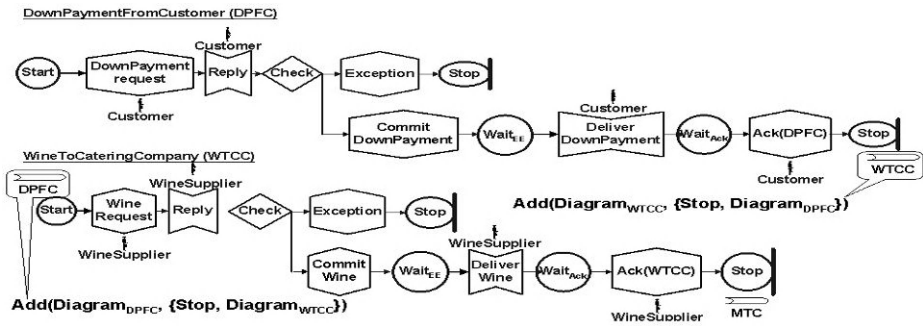


Fig. 14. BML process segment (DownPayFromCustomer and WineToCateringCompany processes)

The relevant two rules that complete inter process communication for the above two inequalities can be listed as below.

if $AEE_i < AEE_j$ then Add(DIAGRAM _i , {Stop, DIAGRAM _j }) Add(DIAGRAM _i , {Wait _{t_E} , DIAGRAM _j })	if $decl(AEE_i) < dir(AEE_j)$ then Add(DIAGRAM _j , {Stop, DIAGRAM _i }) Add(DIAGRAM _j , {Start, DIAGRAM _i })

5 Concluding Remarks

The main contribution of this paper is a set of methodological guidelines that support a designer in moving from a business model to a process model in a systematic way. The approach provides a number of advantages:

Identifying alternatives. The designer assistant helps the designer to identify and evaluate possible design alternatives when building the process model and thereby ensures that no useful alternatives are overlooked.

Traceability and motivation. When inspecting a process model, it is often difficult to understand why a particular solution has been chosen. By building a process model by means of the designer assistant, all design choices as well as their motivations are automatically and explicitly recorded.

Separation of concerns. The approach suggested makes an explicit distinction between the declarative aspects of a business model and the procedural aspects of a process model. This separation of concerns aids a designer in focussing on one problem at a time.

Seamless transition from analysis to realization. Using the designer assistant, the designer starts with a business model and builds successively a process model on its basis. The end point of this activity is a set of diagrams that can be used for communication about the model as well as for actual execution.

In this paper, we have only covered the simplest form of an e-commerce process. Further work is, therefore, needed to handle extensions such as negotiations, breakdowns, cancellations, etc. Furthermore, the scope of the processes could also be extended to handle additional phases in e-commerce, like contact search as in BAT.

References

1. Austin J.L., "*How to do things with words*", Oxford, 1962.
2. Dietz, J.L.G.; Barjis, J.: "Petri Net expressions of DEMO Process Models as a rigid foundation for Requirements Engineering", *The 2nd Int. Conference on Enterprise Information Systems (ICEIS'00)*, 2000.
3. Goldkuhl G (1998) "The six phases of business processes - business communication and the exchange of value", *12th biennial ITS conference (ITS'98)*, Stockholm.
4. Gordijn J., Akkermans J. M. & Vliet J. C.: "What's in an Electronic Business Model? ", *Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th Int. Conference*, Springer-Verlag 2000.
5. Greets G. and McCarthy W. E., "*The Ontological Foundation of REA Enterprise Information Systems*", Michigan State University (revised for journal submission), 2000.
6. Jayaweera P., Johannesson P., & Wohed P.: "From Business Model to Process Patterns in e-commerce", *6th Int. Workshop on the Language-Action Perspective on Communication Modeling* 2001, Montreal, Canada.
7. Johannesson P. and Perjons E.: "Design Principle for Application Integration", *12th Conference on Advanced Information Systems Engineering*, Springer LNCS, 2000.
8. Malone et al.: "Towards a handbook of organizational processes", *MIT eBusiness Process Handbook* <http://ccs.mit.edu/21c/mgtsci/index.htm>, 08th of Nov. 2001.
9. Porter M. E.: "*Competitive Advantages. Creating and Sustaining Superior Performance*" The Free Press 1998.
10. Raul Medina et al.: "The Action Workflow Approach to Workflow Management Technology", *4th Conference on Computer Supported Cooperative Work*, 1992.
11. SDL Standards, <http://www.sdl-forum.org/Publications/Standards.htm>, 08th of Nov. 2001.
12. Searle J.: "*Speech Acts - An Essay in the Philosophy of Language*", Cambridge University Press 1969.
13. Taylor J.: "The Limits of Rationality in Communication Modeling – a Semiotic Reinterpretation of the Concept of "Speech Act"", *3rd Int. Workshop, The Language Action Perspective on Communication Modeling*, 1998.
14. Technical Report, "Mapping Function to Generate BML Process Model", url: <http://www.dsv.su.se/~prasad/html/MapFun.doc>, 08th of Nov. 2001.
15. UN/CEFACT, url: <http://www.unece.org/cefact/>, 08th of Nov. 2001.
16. Weigand H., Moor A. de & Heuvel W-J. van den, "Supporting the evolution of workflow patterns for virtual communities", *33rd Hawaii Int. Conference of System Sciences*, 2000.
17. Wohed, P., "Conceptual Patterns for Reuse in Information Systems Analysis", *12th Int. Conference on Advanced Information Systems - CAiSE 2000*.

Formalising Feasibility and Correctness of Distributed Business Processes

Tomasz Janowski and Adegboyega Ojo*

The United Nations University
International Institute for Software Technology
P.O. Box 3058, Macau
{tj,ao}@iist.unu.edu

Abstract. A distributed business process organizes activities by several enterprises to fulfill a given business goal. The purpose of this paper is to formalise what it means for such a process to be feasible (possible to carry out given the resources delegated for its execution) and for a feasible process to be correct (satisfying a given business goal), using customer-driven manufacturing as a particular, although broadly defined business area. Possible applications are: formal analysis of business processes, providing formal semantics to process modelling languages, and specification and rigorous development of business-support software.

1 Introduction

Business process modelling can have many applications. It can be used to: monitor and control progress of work in an organization; measure and optimise work performance; plan for and predict the effects of operational changes; specify and develop software to further organisational goals, etc. Formal analysis is a particularly good reason to carry out such modelling, as it can disclose problems long before the process is deployed in an organisation. This early detection is especially important for inter-organizational processes where errors, as in all distributed systems, are inherently hard to detect and expensive to correct.

One approach to model a business process is to describe its concrete execution in an enterprise. Typically, such descriptive modelling would be carried out to explore the analogy between business and computing and would concentrate on the mechanics of a process. A prescriptive model, in contrast, describes the purpose of each process. The challenge for business process modelling is, we believe, finding suitable abstractions that can be applied at both levels. By formally relating such descriptive and prescriptive models, we could verify if a process is correct and further develop an engineering approach to design such processes in a rigorous way. This paper is set to contribute to this general goal.

We consider a particular, although broadly defined area of business activity: customer-driven production. Production describes the process of creating goods – tangible products like cars, phones or shovels – produced by assembly from

* On leave from the Computer Science Department, University of Lagos, Nigeria.

sub-products. Production is carried out within production cells, able to store, manufacture and deliver products to its customers. On a certain level, a cell represents formally what is a manufacturing enterprise with its warehouses, shop-floor, product stocks, etc. The behaviour of a cell is driven by the customer orders and how such orders are implemented using production processes.

Our aim is to define formally what it means for a production process to be feasible (possible to carry out given the resources) and for a feasible process to be correct (satisfying a customer order). We adopt the following business model:

1. Sequential production. Each process responds to a customer order which specifies the product to deliver, the number of items and the latest time of delivery. The process operates within a cell which offers the resources for its execution, in terms of product stocks, storage space, and capacity to carry out manufacturing. It describes in detail what operations should be performed and in which sequence. The process is feasible if the cell has enough resources for its execution. A feasible process is correct if its execution satisfies the order: the stocks reach the required volumes within the deadline.
2. Concurrent production. A cell may contain several processes serving individual customer orders, all executed concurrently. In order to resolve conflicts for the shared resources processes are assigned priorities to represent their importance. Feasibility means the cell has enough resources for all such processes, executed concurrently. Correctness means all processes satisfy their corresponding orders, when executed concurrently under a scheduler.
3. Distributed production. Processes running in different cells may satisfy an order in a cooperative way. To utilise the resources that one cell lacks and another has available, they form customer-supplier relations, by receiving, implementing and sending customer orders. Each cell may be running several processes contributing part of the original order. Feasibility means each cell has enough resources for its processes executed concurrently. Correctness means a process satisfies its customer order provided all supplier processes satisfy their own, perhaps executed concurrently with other processes.

The rest of this paper is as follows. Section 2 is about product modelling. Section 3 presents a descriptive production model, and defines what it means for a process to be feasible. Section 4 presents a corresponding prescriptive model. Section 5 defines what it means for a feasible process to be correct, by relating such descriptive and prescriptive models. Section 6 provides some conclusions.

2 Modelling Products

The aim of this section is to provide a definition for products, to be used by both descriptive and prescriptive models.

Products are either atomic, in which case their internal structure is invisible, or are composed of sub-products which are products in their own right. Formally, we introduce the type **Product** to represent products, **Quantity** to represent numbers of products, and **Products** to represent sets of products with quantities. Using the formal specification language RSL [11] we could write:

type

Product,
Quantity = **Nat**,
Products = Product $\xrightarrow{m}$ Quantity

We introduce two functions on the type `Product`: given a product `p`, `size` returns a number `size(p)` to represent the storage requirement of `p` (encoded in some way as `Nat`), and `bill` returns a product map `bill(p)` which represents the immediate sub-products `q` of `p` and how many items of `q` a single item of `p` requires. We constrain those functions in three ways: the value of `size` is always greater than zero, `bill` never returns a map with a zero value and it does not allow for a product which is a sub-product of itself.

type

Size = **Nat**

value

size: Product $\rightarrow$ Size,

bill: Product $\rightarrow$ Products

axiom

$$\forall p: \text{Product} \bullet \text{size}(p) > 0 \wedge$$
$$\sim_{\text{issub}}(p, p) \wedge 0 \notin \text{rng } \text{bill}(p)$$

value

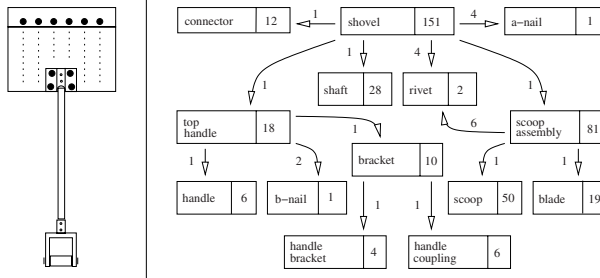
$$\text{issub: Product} \times \text{Product} \rightarrow \mathbf{Bool}$$

Function `issub` takes two products as arguments and returns `true` or `false`, depending if one is a sub-product (immediate or not) of another. We also define some auxiliary functions on product maps: `sum` adds the quantities in the product map, `add` increases the quantity for a given product in the product map, and the overloaded `add` adds two product maps. Here are the declarations:

value

sum: Products $\rightarrow$ Quantity,
 add: Products $\times$ Product $\times$ Quantity $\rightarrow$ Products,
 add: Products $\times$ Products $\rightarrow$ Products

Example 1. Following [14], consider an example product, a snow shovel. The right-hand side of the figure below shows the sub-product graph, which includes the values of the functions **size** (numbers in boxes) and **bill** (arrows and labels). In particular, a shovel has the storage requirement of 151 and it contains one connector, one top handle, one shaft, four rivets, one scoop assembly and four a-type nails. The sub-product graph satisfies the axioms.



3 Descriptive Production Modelling

This section presents a descriptive production model, using the concepts like production cells, production operations and processes. We also define what it means for a process to be feasible given the resources allocated for its execution. Each concept is described and illustrated in the following sections.

3.1 Production Cell

Production is carried out within a production cell; type **Cell**. Each cell provides four kinds of resources: storage, stocks, assembly and transportation. Storage defines the maximum number of products the cell is able to store, represented by the function **storage** from **Cell** to **Size**. Stocks denote the quantity of each product present in the cell, the function **stocks** from **Cell** to **Products**. Assembly describes how the cell can manufacture products from sub-products, using the workstations available in the cell (abstract type **Workstation**). Given a cell and a workstation, function **assemble** returns a map in **Products**: what products can be assembled on this workstation and how many items during a shift. Transportation constrains the maximum number of products the cell can transport between any two workstations within a shift; function **transport**.

type	value
Cell, Workstation	storage: $\text{Cell} \rightarrow \text{Size}$, stocks: $\text{Cell} \rightarrow \text{Products}$, assembly: $\text{Cell} \times \text{Workstation} \rightarrow \text{Products}$, transport: $\text{Cell} \rightarrow ((\text{Workstation} \times \text{Workstation}) \rightarrow \text{Size})$

We extend **stocks** into a total function from **Cell** and **Product** to **Quantity**, which returns the stock of a product, if there is some stock, and otherwise zero.

Such functions are constrained in three ways: (1) the occupancy of the cell must not exceed its storage capacity; (2) each manufacturable product has a non-empty set of sub-products and the quantity manufacturable per-shift is at least one; and (3) no product can be transported from a workstation to itself.

axiom

$$\begin{aligned}
 &\forall c: \text{Cell} \bullet \text{storage}(c) \geq \\
 &\quad \text{sum}([p \mapsto \text{size}(p) * \text{stocks}(c)(p) \mid p: \text{Product} \bullet p \in \mathbf{dom} \text{stocks}(c)]) \wedge \\
 &\quad (\forall w: \text{Workstation} \bullet \text{transport}(c)(w, w) = 0 \wedge \\
 &\quad (\forall p: \text{Product} \bullet p \in \mathbf{dom} \text{assembly}(c, w) \Rightarrow \\
 &\quad \quad \text{bill}(p) \neq [\wedge \text{assembly}(c, w)(p) > 0])
 \end{aligned}$$

Example 2. Consider a cell to produce shovels, **cell**₀. Table below describes the values of **storage**, **stocks**, **assembly** and **transport**. In particular, the cell has the storage capacity of 200000. Stocks include zero shovels, 500 scoop assemblies, etc. The occupancy is 145000. The cell contains three workstations: **ws**₁ can assemble 150 shovels per-shift; **ws**₂ can assemble 100 shovels or 150 top handles; and **ws**₃ can assemble 200 brackets or 150 scoop assemblies. The

transportation system allows to move: 5000 storage units from **ws**₂ to **ws**₁; 15000 units from **ws**₃ to **ws**₁; and 2500 units from **ws**₃ to **ws**₂, all per-shift.

storage	stocks		assembly		transport			
200000	shovel	0	ws ₁			ws ₁	ws ₂	ws ₃
	scoopass	500	shovel	150	ws ₁	0	0	0
	rivet	5000			ws ₂	5000	0	0
	connector	1000	ws ₂		ws ₃	15000	2500	0
	tophandle	500	shovel	100				
	shaft	1000	tophandle	150				
	anail	2000						
	blade	500	ws ₃					
	...	...	bracket	200				
	occupancy	145000	scoopass	150				

We assume that **storage**, **assembly** and **transport** do not change during production, unlike **stocks**. We consider two stock-changing operations: **receive** increments the stock for a given product, provided the cell has enough space, **deliver** decrements the stock for a product, provided the stock is sufficient. They are extended into the corresponding functions on product maps.

value

receive: Product $\times$ Quantity $\times$ Cell $\xrightarrow{\sim}$ Cell
 receive(*p*, *n*, *c*) **as** *c'* **post** stocks(*c'*, *p*) = stocks(*c*, *p*) + *n* $\wedge$...
 pre enough_receive(*p*, *n*, *c*),
 deliver: Product $\times$ Quantity $\times$ Cell $\xrightarrow{\sim}$ Cell
 deliver(*p*, *n*, *c*) **as** *c'* **post** stocks(*c'*, *p*) = stocks(*c*, *p*) - *n* $\wedge$...
 pre enough_deliver(*p*, *n*, *c*)

Example 3. Consider the operation to receive 2000 a-nails and 2000 rivets by **cell**₀, Example 2. The additional products occupy 6000 storage units, therefore can be still accommodated by the cell. The resulting cell is called **cell**₁.

3.2 Production Operations

Production is carried out by the allocation of work to workstations. Throughout the shift, each workstation is programmed to assemble a number of items of a given product from its sub-products. Sub-products come either from the warehouse or from other workstations. For each workstation we decide which part of the volume is forwarded to which workstations for further processing; the rest is returned to the warehouse. The type **Work** describes the allocation of work to a workstation. A well-formed **work** is such that the total volume produced and the volumes forwarded are all non-zero, and the sum of the quantities forwarded does not exceed the volume produced. The type **Operation** is a map from workstations to **Work**. A well-formed operation is such that: forwarding of products occurs only between the workstations in domain of the operation; a workstation

is only forwarded sub-products of the product it currently manufactures; and the total number of items received by the workstation from others does not exceed the number needed to carry out production in the current shift.

type

Work:: volume: Quantity

product: Product forward: Workstation $\overrightarrow{\text{m}}$ Quantity,

Operation = Workstation $\overrightarrow{\text{m}}$ Work

Example 4. Consider the operation **op** on the right. It assigns the workstations **ws₁**, **ws₂** and **ws₃** the tasks to assemble 150 shovels, 150 top handles and 150 scoop assemblies respectively. The entire volume of shovels is sent to the inventory. Half of the volume of top handles and scoop assemblies is forwarded to **ws₁** to produce shovels and half is sent to the inventory.

	ws ₁	ws ₂	ws ₃
product	shovel	tophandle	scoopass
volume	150	150	150
forward		ws ₁ 75	ws ₁ 75

3.3 Production Operations: Feasibility

The function **enough** from **Operation** and **Cell** to **Bool** decides if a given cell has enough resources to carry out a given operation: (1) each workstation is able to carry out the work assigned to it; (2) the cell has enough stocks to satisfy the requirement for sub-products from all work assignments (minus the sub-products received from other workstations); and (3) the cell has enough space to accommodate manufactured products. Here is the definition:

value

enough: Operation $\times$ Cell $\rightarrow$ **Bool**

enough(op, c) $\equiv$

enough_receive(to_stock(op), c) $\wedge$

enough_deliver(from_stock(op), c) $\wedge$

($\forall$ ws: Workstation $\bullet$ ws $\in$ **dom** op $\Rightarrow$ enough(op, ws, c))

We applied three auxiliary functions. The overloaded function **enough** checks whether each workstation is able to carry out its work assignment: it can assemble the product; the volume can be obtained within a shift; and the cell has enough capacity to transport the required quantities from this workstation to others. The function **from_stock** decides how many sub-products the operation needs from the stock to carry out all work assignments. The function **to_stock** accumulates the results produced by each individual workstation: the product and the volume assigned to be produced by the workstation, minus the amounts sent to other workstations for further processing.

Example 5. The operation **op** (Example 4) is feasible for its execution on **cell₁** (Example 3): each product is in the **assembly** map of the corresponding workstation, the volumes do not exceed what the workstations can assemble within a shift, there is enough transportation capacity and enough sub-products in stock.

3.4 Production Operations: Execution

Consider a cell c and an operation op which is feasible for execution on c . The function **exec** describes this execution, which results in a modified cell $\mathbf{exec}(op, c)$. The cell has its stocks increased according to $\mathbf{to\_stock}(op)$ – the products manufactured but not consumed by the operation, and also decreased according to $\mathbf{from\_stock}(op)$ – the products consumed by op . In order to modify the cell, **exec** applies the functions **receive** and **deliver**.

value

$\mathbf{exec}: \text{Operation} \times \text{Cell} \xrightarrow{\sim} \text{Cell}$

$\mathbf{exec}(op, c) \equiv \mathbf{deliver}(\mathbf{from\_stock}(op), \mathbf{receive}(\mathbf{to\_stock}(op), c))$

pre enough(op, c)

Example 6. Consider the execution of op (Example 4) on $\mathbf{cell}_1$ (Example 3). The result, $\mathbf{cell}_2$, is obtained by increasing the stock for shovel and decreasing the stocks for all sub-products of shovel, top handle and scoop assembly. The stocks for top handle and scoop assembly remain unchanged, as the entire volume needed to produce 150 shovels is obtained by production.

3.5 Production Operations: Concurrency

For two operations to be executed concurrently, the workstations in both operations must be assigned to produce the same product. This is captured by the function **compatible**. Given two compatible operations, their composition is done by the function **compose** which takes all work assignments from the non-shared workstations without change and combines the assignments for the shared workstations (adds the volumes produced and forwarded).

value

$\mathbf{compatible} : \text{Operation} \times \text{Operation} \rightarrow \mathbf{Bool}$,

$\mathbf{compose}: \text{Operation} \times \text{Operation} \xrightarrow{\sim} \text{Operation}$

Given a cell c and two compatible operations op_1 and op_2 , their feasibility on c means feasibility of the operation $\mathbf{compose}(op_1, op_2)$. If feasible, concurrent execution of op_1 and op_2 is realised as the usual execution of $\mathbf{compose}(op_1, op_2)$.

value

$\mathbf{enough}: \text{Operation} \times \text{Operation} \times \text{Cell} \xrightarrow{\sim} \mathbf{Bool}$

$\mathbf{enough}(op_1, op_2, c) \equiv$

$\mathbf{enough}(\mathbf{compose}(op_1, op_2), c)$ **pre** $\mathbf{compatible}(op_1, op_2)$,

$\mathbf{exec}: \text{Operation} \times \text{Operation} \times \text{Cell} \xrightarrow{\sim} \text{Cell}$

$\mathbf{exec}(op_1, op_2, c) \equiv \mathbf{exec}(\mathbf{compose}(op_1, op_2), c)$

pre $\mathbf{compatible}(op_1, op_2) \wedge \mathbf{enough}(op_1, op_2, c)$

Example 7. Consider the operations op_1 and op_2 below. The operations are compatible. They manufacture together 150 shovels, producing the required quantities of top handles and scoop assemblies. Their concurrent composition $compose(op_1, op_2)$ is the operation op (Example 4).

	ws ₁	ws ₂
product	shovel	tophandle
volume	75	150
forward		ws ₁ 75

	ws ₁	ws ₃
product	shovel	scoopass
volume	75	150
forward		ws ₁ 75

3.6 Production Processes

A production process is a sequence of production operations, defined as the type **Process**. Functions **enough** and **exec** check if a given cell has enough resources for a process (the process is feasible) and if so execute this process, respectively.

type

Process = Operation*

value

enough: Process $\times$ Cell $\rightarrow$ **Bool**

enough(p, c) $\equiv p = \langle \rangle \vee \text{enough}(\mathbf{hd} \ p, c) \wedge$

enough($\mathbf{tl} \ p, \text{exec}(\mathbf{hd} \ p, c)$),

exec: Process $\times$ Cell $\leadsto$ Cell

exec(p, c) $\equiv$ **if** $p = \langle \rangle$ **then** c **else**

exec($\mathbf{tl} \ p, \text{exec}(\mathbf{hd} \ p, c)$) **end pre** enough(p, c)

Example 8. Consider the process p on the right, which purpose is to produce 1000 shovels. The process is feasible for execution on $cell_1$ (Example 3). After the execution, the final state includes 1000 shovels but zero quantities of all other products.

Given the processes p_1 and p_2 , their concurrent execution means to execute concurrently $\mathbf{hd} \ p_1$ and $\mathbf{hd} \ p_2$ and then continue concurrently with $\mathbf{tl} \ p_1$ and $\mathbf{tl} \ p_2$. If the execution

of $\mathbf{hd} \ p_1$ and $\mathbf{hd} \ p_2$ is not possible (the operations are incompatible or the cell has insufficient resources, or a process is empty), $\mathbf{hd} \ p_1$ will be executed and if $\mathbf{hd} \ p_1$ is not possible, $\mathbf{hd} \ p_2$ will be executed. The function **enough** decides if the cell c has enough resources to allow at least one of p_1 or p_2 to make progress. We express this concurrent execution of p_1 and p_2 as sequential execution of the process $compose(p_1, p_2, c)$; **compose** takes **enough** as a precondition.

value

enough: Process $\times$ Process $\times$ Cell $\rightarrow$ **Bool**,

compose: Process $\times$ Process $\times$ Cell $\leadsto$ Process

ws ₁	ws ₂	ws ₃
shovel	tophandle	scoopass
150	150	150
	ws ₁ 75	ws ₁ 75

shovel	tophandle	bracket
150	150	200
	ws ₁ 150	ws ₂ 150

shovel	shovel	scoopass
150	100	150
		ws ₁ 150

shovel	tophandle	scoopass
150	150	150
	ws ₁ 75	ws ₁ 75

shovel	tophandle	bracket
150	50	50
	ws ₁ 50	ws ₂ 50

shovel	shovel	scoopass
50	100	50
		ws ₁ 50

Example 9. Consider how to divide the process p (Example 8) into concurrent processes p_1 and p_2 such that $p = \text{compose}(p_1, p_2, \text{cell}_1)$ and each process is responsible for manufacturing 500 shovels on cell_1 (Example 3). However, p_1 must not use the stock of top handles but manufacture them from sub-products and p_2 must not use the stock of scoop assemblies but manufacture them.

4 Prescriptive Production Modelling

Section 3 introduced a concrete model for describing production, including production resources, operations on them, what it means for such operations to be feasible for execution and what is the execution effect in this case. Here we look at such operations from the point of view of their purpose, not their execution.

The purpose of each production cell is to continuously satisfy the needs of its customers, expressed in terms of products, quantities and deadlines. This “mission statement” translates down to the goals of individual processes, each designed to respond to a customer order. The type **Order** is defined as:

type

Order == $\text{produce}(\text{Products}) \mid \text{deliver}(\text{Products}, \text{Order}) \mid$
 $\text{receive}(\text{Products}, \text{Order}) \mid \text{and}(\text{Order}, \text{Order}) \mid \text{deadline}(\text{Nat}, \text{Order})$

The order $\text{produce}(\text{ps})$ describes the set of products the customer needs and how many items of each such product. $\text{deliver}(\text{ps}, o)$ means to satisfy the order o after delivering the products ps ; an implementations of o must not rely on ps to still exist in the cell. $\text{receive}(\text{ps}, o)$ means to satisfy the order o after receiving the products ps ; an implementation of o may rely on those additional products. $\text{and}(o_1, o_2)$ means to satisfy the orders o_1 and o_2 simultaneously, where neither depends on the results produced by the other. $\text{deadline}(n, o)$ means to satisfy the order o within the number of shifts described by n .

Example 10. Consider a customer order to produce 1000 shovels to be ready within six shifts from the moment a production cell receives 2000 rivets and 2000 a-nails. Let $\text{ps}_1 = [\text{shovel}_1 \rightarrow 1000]$ and $\text{ps}_2 = [\text{river}_1 \rightarrow 2000, \text{anail}_1 \rightarrow 2000]$. The order is described as $\text{deadline}(6, \text{receive}(\text{ps}_2, \text{produce}(\text{ps}_1)))$.

5 Relating Descriptive and Prescriptive Models

The aim of this section is to formally relate descriptive and prescriptive models defined in Sections 3 and 4. Consider a production cell c , a process p which is feasible for its execution on c , and a customer order o . Function **satisfies** defines what it means for p to satisfy o when executed on c .

value

satisfies: $\text{Process} \times \text{Order} \times \text{Cell} \rightarrow \text{Bool}$

We define this function for different kinds of customer orders. Consider the order **produce**(**ps**): after executing **p** on **c**, the resulting levels of stocks must satisfy the quantities in the product map **ps**:

$$\text{satisfies}(\mathbf{p}, \text{produce}(\mathbf{ps}), \mathbf{c}) = (\forall r: \text{Product} \bullet r \in \mathbf{dom} \mathbf{ps} \Rightarrow \text{stocks}(\text{exec}(\mathbf{p}, \mathbf{c}), r) \geq \mathbf{ps}(r))$$

Example 11. Consider the process **p** (Example 8), the cell **cell₁** (Example 3) and the customer order **produce**(**ps₁**) to produce 1000 shovels (Example 10). The process satisfies **produce**(**ps₁**) when executed on **cell₁**.

Process **p** satisfies an order **deliver**(**ps**, **o**) when executed on **c** iff **p** can manufacture **ps** on **c** and then satisfy **o** without relying on those manufactured products. Here is the axiom expressing this:

$$\begin{aligned} \text{satisfies}(\mathbf{p}, \text{deliver}(\mathbf{ps}, \mathbf{o}), \mathbf{c}) = & (\exists \mathbf{p1}, \mathbf{p2}: \text{Process} \bullet \\ & \mathbf{p} = \mathbf{p1} \wedge \mathbf{p2} \wedge \text{enough}(\mathbf{p1}, \mathbf{c}) \wedge \text{satisfies}(\mathbf{p1}, \text{produce}(\mathbf{ps}), \mathbf{c}) \wedge \\ & \mathbf{let} \mathbf{c1} = \text{exec}(\mathbf{p1}, \mathbf{c}), \mathbf{c2} = \text{deliver}(\mathbf{ps}, \mathbf{c1}) \\ & \mathbf{in} \text{enough}(\mathbf{p2}, \mathbf{c2}) \wedge \text{satisfies}(\mathbf{p2}, \mathbf{o}, \mathbf{c2}) \mathbf{end}) \end{aligned}$$

Process **p** satisfies an order **and**(**o₁**, **o₂**) on **c** iff it is a concurrent composition of some **p₁** and **p₂** such that **p₁** satisfies **o₁** in the presence of **p₂** executing concurrently (**p₁** is feasible for **c₂** obtained by removing from **c** the products that **p₂** needs for its execution, and it satisfies **o₁** on **c₂**) and vice versa, **p₂** satisfies **o₂** in the presence of **p₁** executing concurrently. This way we ensure that the processes do not interfere with each other's goals. Here is the axiom:

$$\begin{aligned} \text{satisfies}(\mathbf{p}, \text{and}(\mathbf{o1}, \mathbf{o2}), \mathbf{c}) = & (\exists \mathbf{p1}, \mathbf{p2}: \text{Process} \bullet \\ & \text{enough}(\mathbf{p1}, \mathbf{p2}, \mathbf{c}) \wedge \mathbf{p} = \text{compose}(\mathbf{p1}, \mathbf{p2}, \mathbf{c}) \wedge \\ & \mathbf{let} \mathbf{c2} = \text{deliver}(\text{from\_stock}(\mathbf{p2}), \mathbf{c}) \mathbf{in} \text{satisfies}(\mathbf{p1}, \mathbf{o1}, \mathbf{c2}) \mathbf{end} \wedge \\ & \mathbf{let} \mathbf{c1} = \text{deliver}(\text{from\_stock}(\mathbf{p1}), \mathbf{c}) \mathbf{in} \text{satisfies}(\mathbf{p2}, \mathbf{o2}, \mathbf{c1}) \mathbf{end}) \end{aligned}$$

Example 12. Consider the process **p** (Example 8), the cell **cell₁** (Example 3), and the customer order **and**(**produce**(**ps**), **produce**(**ps**)) to produce 500 shovels concurrently with another 500. The process is feasible for **cell₁** and it satisfies the order, using the decomposition into **p₁** and **p₂** (Example 9).

Process **p** satisfies the order **receive**(**ps**, **o**) when executed on **c** iff it satisfies **o**, after **c** has received the products **ps**. The assumption is that **c** must have enough space to accommodate the products **ps**. Here is the axiom:

$$\begin{aligned} \text{satisfies}(\mathbf{p}, \text{receive}(\mathbf{ps}, \mathbf{o}), \mathbf{c}) = & \\ & (\text{enough\_receive}(\mathbf{ps}, \mathbf{c}) \wedge \text{satisfies}(\mathbf{p}, \mathbf{o}, \text{receive}(\mathbf{ps}, \mathbf{c}))) \end{aligned}$$

Example 13. Consider the process **p** (Example 8), the cell **cell₀** (Example 2), and the customer order **receive**(**ps₂**, **produce**(**ps₁**)) (Example 10). The order requires the production of 1000 shovels provided the cell first receives 2000 rivets and 2000 a-nails. The process indeed satisfies this order on **cell₀**.

Process p satisfies the order $\text{deadline}(n, o)$ on c iff it satisfies o and completes its execution within n shifts, after receiving all external products.

$$\text{satisfies}(p, \text{deadline}(n, o), c) = (\text{satisfies}(p, o, c) \wedge \text{len } p \leq n)$$

Example 14. Consider the process p (Example 8), the cell cell_0 (Example 2), and the order $\text{deadline}(6, \text{receive}(\text{ps}_2, \text{produce}(\text{ps}_1)))$ (Example 10). The order requires the production of 1000 shovels within 6 shifts, after receiving 2000 rivets and 2000 a-nails. The process indeed satisfies this order on cell_0 .

6 Conclusions

Being able to assert correctness of a computer program by mathematical reasoning, rather than testing is the cornerstone of formally-based software development. A number of programming theories now exist, with their own definitions and the means to verify program correctness. Such theories express correctness in terms of program variables, relations between inputs and outputs, sequences of state-changes, possible interactions with environment, etc. Program feasibility is either a non-issue (data is easy to replicate), abstracted away (assuming enough resources exist) or addressed by static or dynamic scheduling.

This chapter is an attempt at defining correctness for business processes, using the concepts from the business domain to express what this correctness really means. We focused on a particular business domain: manufacturing of discrete-parts products, for which we defined models to describe what is a business process, what are the resources the process needs for its execution, when there are enough resources and what the execution really means. We expressed such definitions formally, using what we considered a “minimum” number of modelling concepts. In addition to this descriptive model, we also defined a corresponding prescriptive model to capture the purpose of each process. Correctness was then defined as a relation between such descriptive and prescriptive models.

There is a vast literature concerning business process modelling, from the perspectives of management, operations research, production research, industrial engineering, software development, information systems, artificial intelligence, etc. For instance, [1] promotes a process-oriented view of an enterprise in order to best utilise the promise of enterprise systems to further organisational goals. [6] describes an informal graphical notation to carry out process analysis by simulation. Well-known examples of industrial frameworks are ARIS [10] and CIMOSA [9]. [13] contains an overview of such frameworks from the point of view of enterprise modelling and integration. [5] describes the need to base enterprise description languages on formal semantics and [8] contains an evaluation of the existing languages and tools. [4] contains an operations management perspective on modelling of business process flows, allowing in particular quantitative analyses of process performance. Among formally-based approaches, [3] describes a modelling language allowing various kinds of analyses by translation into different semantic models. Petri Nets is one of the most popular formal approaches, in particular for modelling manufacturing processes [2], and for workflow systems

[12,15]. The latter papers consider verification of workflows but express properties like absence of control-flow errors, soundness or termination in technical, not business terms. For us, such properties would be used to establish that a process is well-formed, a necessary but insufficient condition for correctness.

Future work in this area has several possible directions. First, to further develop the model: add more structure to process definitions; support different ways to schedule processes; make explicit the semantics of the prescriptive model, instead of relying on the descriptive model; derive algebraic laws about orders; etc. Second, to automate the construction of a process to satisfy a given customer order [7]. Third, to describe the model with XML and use this representation to develop an implementation. Finally, to build some real-life case studies.

Acknowledgments. The authors would like to thank Chris George for useful comments on the contents and presentation of this report.

References

1. T.H. Davenport. *Mission Critical, Realising the Promise of Enterprise Systems*. Harvard Business School Press, 2000.
2. A.A. Desrochers and R.Y. Al-Jaar. *Applications of Petri Nets in Manufacturing Systems*. IEEE Press, 1995.
3. H. Eertink et al. A Business Process Design Language. In *Formal Methods Congress*, volume 1708 of *LNCS*, pages 76–95. Springer, 1999.
4. R. Anupindi et al. *Managing Business Process Flows*. Prentice Hall, 1999.
5. J. Goossenaerts, M. Gruninger, J.G. Nell, M. Petit, and F. Vernadat. Formal Semantics of Enterprise Models. In K. Kosanke, editor, *Proceedings of ICEIMT'97, Berlin*. Springer Verlag, 1997.
6. G.A. Hansen. *Automating Business Process Reengineering*. Prentice Hall, 1997.
7. T. Janowski. Distributed production with specification-generated processes. In *4th IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems in Production and Transportation (BASYS00)*. Kluwer, 2000.
8. W. Janssen, H. Jonkers, and J.P.C. Verhoosel. What makes business processes special? An evaluation framework for modelling languages and tools in Business Process Redesign. In *2nd IFIP Workshop on Evaluation of Modelling Methods in Systems Analysis and Design, Barcelona*, 1997.
9. K. Kosanke. *CIMOSA: Open System Architecture for CIM*. Springer Verlag, 1993.
10. A.W. Scheer. *ARIS – Business Process Frameworks*. Springer, 1998.
11. The RAISE Method Group. *The RAISE Specification Language*. Pr. Hall, 1992.
12. W.M.P. van der Aalst. Workflow Verification: Finding Control-Flow Errors using Petri-net-based Techniques. In *Business Process Management: Models, Techniques, and Empirical Studies*, volume 1806 of *LNCS*, pages 161–183. Springer, 2000.
13. F. Vernadat. *Enterprise Modeling and Integration*. Chapman and Hall, 1996.
14. T.E. Vollmann. *Manufacturing, Planning and Control Systems*. Irwin, 1992.
15. M. Voorhoeve. Compositional Modeling and Verification of Workflow Processes. In *Business Process Management: Models, Techniques, and Empirical Studies*, volume 1806 of *LNCS*, pages 184–200. Springer, 2000.

Modeling Products for Versatile E-commerce Platforms – Essential Requirements and Generic Design Alternatives

Ulrich Frank

University of Koblenz, Rheinau 1, 56075 Koblenz
`ulrich.frank@uni-koblenz.de`

Abstract. Electronic commerce (e-commerce) covers a wide range of business transactions. Usually, a transactions includes products, i.e. goods or services. Often, e-commerce websites are intended to offer a large variety of products, most of which are not fully recognized at the time when the system is constructed. Nevertheless, there is need to describe these products in an elaborated way in order to support prospective customers in finding adequate products. Since new product types may have to be registered on a daily basis, changing a data base schema or program code is no satisfactory option. Instead there is the need to register new product types during the run time of a system. In addition, the representation of product variants and the construction of individual product configurations should be possible. Current approaches to represent products do not fulfill these requirements to a satisfactory degree. The design of appropriate conceptual models requires to thoroughly differentiate between various levels of abstraction. This includes decisions to be made between instantiation, specialization and "uses" relationships. Against this background, the paper presents three prototypical design alternatives for versatile product representations. They vary in terms of flexibility, reusability and ease of use. The work presented in this paper resulted from the design of a reference model for e-commerce platforms.

1 Introduction

While electronic data interchange (EDI) is restricted to the transmission of data within traditional business models, e-commerce allows for entirely new ways of doing business. Among other things, we witness the rise of new kinds of intermediaries that combine trading and brokerage with purely internet based, non-personal acquisition and sales, e.g. companies that offer platforms for electronic auctions or electronic marketplaces in general. We use the term "e-commerce platform" as an abstraction of information systems that support this variety of actual internet businesses. Unlike traditional inventory management, an e-commerce platform is aimed at covering the initiation and performing of business transactions on the Internet between participants that possibly do not know each other. Companies that run e-commerce platforms face a number of remarkable challenges, including the development of lasting customer relationships, the

establishment of a competitive profile and becoming well known in relevant markets with a reasonable advertising budget. Additionally, a pivotal success factor for e-commerce platforms to perform economically is a powerful software architecture. A couple of years ago, a few fancy web pages, and impressive sales forecasts, were sufficient to qualify as a serious e-commerce contender - no matter whether there was only a quick and dirty implementation under the hood. Times have changed. For e-commerce platforms to be enablers of more efficient business transactions, they have to offer more than allowing customers to submit orders through web browsers. In general, they should promote the automation of business processes: even when e-commerce seems to focus at first sight on the generation of business over the Internet (and to which some current offers actually limit themselves), economic reasons suggest an integration and extensive automation of all involved processes, such as procurement, logistics and financial transactions. Also, an e-commerce platform should allow for representing a wide range of products. This is due to the fact that the diversity of products offered over e-commerce websites on the Internet is sometimes considerable; in some cases - for example in auction platforms - it is almost unlimited. In contrast to department stores or mail order companies that also often offer a wide array of products, the range cannot be planned in the medium term. Instead, it must be possible to make diverse, almost totally unpredictable changes to the product range on a daily basis; today, leather seats; tomorrow, helicopters. In order to support a tight integration with other systems in the supply chain, products must be described meaningfully with regard to corresponding use cases. Therefore, it is all about conceptualizing objects of which one has, up to that point, no knowledge. This paper is aimed at conceptual solutions to overcome this seeming paradox.

2 Requirements and State of the Art Solutions

To solve the problem outlined above, we need appropriate conceptual models of products. A conceptual product model does not only guide the implementation of software, it is also a blueprint for the specification of communication interfaces. Before we consider current approaches to represent products in information systems, we will have a more detailed look at the requirements a competitive e-commerce platform should fulfill.

1. In principle, it should be possible to represent any products (or more exactly: types of products).
2. The registration of new product types should not require a change of the program code or of the database schema, because in view of the general availability of the system and the frequent appearance of new product types this would not be acceptable.

Other requirements emphasize the customers' viewpoint:

3. The objects stored on the basis of conceptual product models should support the customer when searching for suitable products.

4. With this, the description of products from the point of view of the customers should have a sufficient content and be fairly set out for all their various needs in a detailed and clear form. This requires, for instance, that both a customer searching a particular type of furniture and another one searching for an excavator should be able to specify relevant features. Common search engines are usually not satisfactory. They operate on full text representations that do not include product descriptions as semantic structures. For example, imagine to search for a dining-table with a glass top and mahogany table-legs. While it would not be possible to express exactly what you are looking for, a search engine would retrieve pages that contain both glass and mahogany tables but maybe none that satisfies the relevant search criteria.

Besides that, there are a series of requirements that are connected to the automated use and care of product data:

5. The product descriptions should allow for the collection, over a period of time, of relevant marketing knowledge about the purchasing behavior of customers. This requires the analysis of buying preferences of customers - grouped by demographic variables - with respect to specific features of products.
6. The system should extensively shut out the registration of meaningless product descriptions. That implies to define products types that restrict the possible descriptions of corresponding instances.

The same is valid for the registration of customer orders:

7. It should be possible to represent product variants as such, since in this way not only can the analysis of purchasing habits mentioned above be supported, but also redundancy of data registration and use can be avoided: Features that have been described for a product type already do not have to be described again when it comes to specify corresponding variants.
8. The customer should be in the position to specify individual configurations, e.g. by assigning equipment to a car. To avoid orders that cannot be satisfied, it is essential to allow for correct configurations only.
9. Different forms of pricing and price assigning (for product types and single products) should be possible.
10. The product descriptions should be versatile enough to satisfy diverse communication interfaces (e.g. to customers, suppliers, banks, and logistical partners).

The representation of products is subject of a number of EDI protocols, such as UN-EDIFACT. Typically, these protocols are aimed at the specification of business documents like orders, invoices etc. They do, however, allow for the use of product identifiers only, thereby relying on the participating parties to share a common understanding of these identifiers. Current standardization initiatives that are aimed at fostering e-commerce transactions, like 'Open Buying on the Internet' [7] or RosettaNet [8], also do without concepts of products. Instead, the XML document types they define, include tags for product identifiers only. This

is the case, too, for document types that are propagated by companies which operate marketplaces on the Internet ([1], [2]). Probably the most common approach to describe product types are data structures that include features that all product types have in common, like 'Name', 'Description', 'Price', 'Picture' etc. The differentiation between various product types follows solely through specific initialization of the generic features. While such flat descriptions are very flexible in the sense that they allow for the representation of any kind of product without the need for changing code (requirements 1, 2), their support for searching products is limited to text retrieval (requirements 3, 4). At the same time, it must be taken into account that a product concept of this kind does not support the differentiated analysis of consumer behavior, since the product types that are offered cannot be distinguished conceptually (requirement 5). Requirement 6 marks an inherent weakness of this concept, since in conceptual terms almost all nonsensical product descriptions are possible. Therefore it jeopardizes the integrity of a system. Specifying product variants is not feasible on a conceptual level since it is not possible to show which characteristics distinguish one variant from another (requirement 7). For similar reasons it is not possible to represent product configurations (requirement 8). Information on pricing and for logistical purposes (requirements 9, 10) could be added. Class hierarchies, where each class represents a specific product type, would allow for more elaborate descriptions of product types. For our purpose, however, it is not satisfactory because the introduction of a new product type would require to define a new specialized class, hence the modification of code (requirements 1, 2). There are a few approaches that allow for more meaningful, yet flexible definition of products. Vendors of e-commerce software, like BEA or Intershop, supplement flat descriptions of product types with entity types that are in fact meta types. They allow for the specification of additional attribute types (www.bea.com, www.intershop.com). BMEcat, an intended XML based standard protocol for B2B transactions, uses a similar approach ([5], [9]). While all of these approaches allow to describe new product types without changing code, none of them allows for the specification of configurations or variants. Also, their focus is mainly on the definition of exchange protocols not on providing for a conceptual foundation of corresponding information systems. Also, none of these approaches is based on a thorough differentiation of the levels of abstraction to be taken into account.

3 Levels of Abstraction

To prepare for the representation of products, it is helpful to clarify the levels of abstraction one has to deal with - and to contrast them with the levels of abstraction that are available in state of the art software architectures.

3.1 Instantiation, Specialization, and Re-use

The definition of new product types requires an appropriate language. In order to avoid definitions that are not compliant with an acceptable notion of product, the "language" can be restricted to a meta model that defines the set of

all syntactically valid product type definitions. The meta model itself could be specified using the ER model or the UML. A simple meta model of this kind could, for instance, consist of two entity types as illustrated in fig. 1. Sometimes, it will not be satisfactory to define a product type from scratch by instantiating it from a meta model. Instead, it may be more appealing to specialize it from an existing product type. Fig. 1 includes an example of a specialized product type. Notice, that in reality there is no fixed limit to the number of levels. Specialization seems to be well suited for any kind of 'is a' relationships. However, it recommends a second thought with respect to the semantics of specialization or inheritance. While in natural language, there is no unique concept of specialization, one interpretation seems to be prevailing. It is based on an extensional notion of a class, which defines a class as a set of instances. Hence, a specialized class is a subset of its superclass. Database theory usually suggests a similar notion of specialization. This is, however, different with common object-oriented programming language that are usually the implementation language of choice. They feature an intensional notion of a class where a class is defined by a certain structure. A specialized class inherits this structure. Different from the extensional concept, an object can be an instance of one and only one class. Applying the intensional concept to product types reveals yet another challenge. Sometimes, the instances of a specialized type may have features with the same state as the instances of the corresponding supertype. Take, for instance, a special edition of a certain type of TV set. Its features (screen size, weight, etc.) have the same value as all the other instances of the type it is specialized from. In addition to these, it includes a computer system that allows for internet access. It is not possible to express this "re-use" of state through specialization that is based on an intensional concept of a class: you would have to instantiate a new instance of the specialized class without any inherited state (for a detailed discussion see [4]). In fig. 1, the term 'cloned' is used to denote this kind of relationship. For many products, there is no need to describe product instances. Sometimes, however, a customer may want to pick a particular instance. This is especially the case for used products. When it comes, for example, to consider the price of a used car, its particular state has to be taken into account. At first sight, this would suggest to represent particular products as instances of a certain product type. However, similar to the specialization into limited edition discussed above, instantiation is not satisfactory. Many of a used car's features will have the same state as all cars of the corresponding type. Using a class (or a type respectively) and its instances does not allow to express this information: Firstly, a class does not allow for storing states that apply to all instances (that would require meta classes). Secondly, the state of an instantiated object is on a different level of abstraction than the state of its class.

3.2 Layers in Common Software Architectures

So far, we have seen that there are many levels of abstraction that have to be taken into account with respect to the representation of products. Also, the relationships between the various levels cannot always be clearly classified as

Level of Abstraction	Example								
Meta language to define product types	<table><tr><td>Product_Type</td><td></td><td></td><td>Feature_Type</td></tr><tr><td>name String</td><td>1..1</td><td>0..*</td><td>name String type String</td></tr></table>	Product_Type			Feature_Type	name String	1..1	0..*	name String type String
Product_Type			Feature_Type						
name String	1..1	0..*	name String type String						
Type product type	<table><tr><td>TV</td></tr><tr><td>make String screenSize Real frequency Integer</td></tr></table>	TV	make String screenSize Real frequency Integer						
TV									
make String screenSize Real frequency Integer									
Initialized Type product type with initialized features	<table><tr><td>TV</td></tr><tr><td>make s\$onyo screenSize 50 frequency 100</td></tr></table>	TV	make s\$onyo screenSize 50 frequency 100						
TV									
make s\$onyo screenSize 50 frequency 100									
Specialized Type product type specialized from product type	<table><tr><td>Internet-TV</td></tr><tr><td>make String screenSize Real frequency Integer OS String</td></tr></table>	Internet-TV	make String screenSize Real frequency Integer OS String						
Internet-TV									
make String screenSize Real frequency Integer OS String									
Cloned Type product type specialized from initialized product type	<table><tr><td>TV</td></tr><tr><td>make s\$onyo screenSize 50 frequency 100 model s\$-90a</td></tr></table>	TV	make s\$onyo screenSize 50 frequency 100 model s\$-90a						
TV									
make s\$onyo screenSize 50 frequency 100 model s\$-90a									
Particular Product Instance particular instance	<table><tr><td>TV</td></tr><tr><td>make s\$onyo... serial s\$-9851ka</td></tr></table>	TV	make s\$onyo... serial s\$-9851ka						
TV									
make s\$onyo... serial s\$-9851ka									

Fig. 1. Levels of abstraction concerning the representation of products

instantiation or specialization. To make things even worse, we have to consider the limitations of common software architectures to represent the abstractions we have identified so far. Usually there are two main levels of abstraction in addition to the implicit language layer. The "meta layer" allows for the definition of concepts such as classes or relation types. The "object layer" serves to store the corresponding instances. Avoiding the modification of code for the purpose of introducing a new product type implies to reserve the meta layer for the definition of a meta model of product types. Product types would then actually be instances on the object layer. This has two implications. Firstly, there is no way to directly use specialization relationships between product types because they can be applied to classes only and not to instances. Secondly, it is not possible to instantiate objects representing particular product instances from product types because an instance cannot be instantiated from an instance. Our brief discussion of the abstractions to be taken into account as well as the concepts available in state of the art software architectures shows that a versatile representation has to face two essential challenges. On the one hand, it is necessary to clarify the levels of abstraction and the corresponding levels of re-use (re-use of concept, re-use of state) that are needed to represent a certain population of products. On the other hand, these abstractions have to be reconstructed using the restricted set of concepts provided with common software development environments.

4 Three Prototypical Design Alternatives

It would be nice if there was one representation of products that would fulfill all requirements (see 2) to a satisfactory degree. However, our work on the design of an object-oriented reference model for e-commerce platforms suggests a differentiated approach. Goals like flexibility and re-usability are in part conflicting and come with additional trade-offs like compromising integrity and ease of (re-) use. Depending on the products to be represented for an e-commerce platform, the relevance of these goals varies. Therefore we introduce three prototypical product models each of which is suited for a set of products that meets certain conditions.

4.1 Emphasis on Versatile Definitions of Product Types and Configurations

Some product types offered on e-commerce platforms allow for individual configurations or come in different variants. The object model shown in fig. 2 allows for a fairly flexible specification of configuration options. At the same time it is limited with respect to the re-use of product types (conceptual re-use) and instantiated type features (instance level re-use). Hence, it is suited for domains where similarity of product types is of no concern - because product types do not have features in common or it would be too expensive to consider similarity when introducing a new product type. To take advantage of the additional flexibility offered by meta concepts within common architectures, it is necessary to overload layers with more than one abstraction. Some of the concepts shown in fig. 2 are part of a product meta model (e.g. `Standard_Feature_Type`, `Optional_Feature_Type`). Others are directly related to certain product types (e.g. `Feature_Value`, `Configuration_Type`). As a consequence, the corresponding instances cover two levels of abstraction, too. This model is valid for all conceivable product types. `Product_Type` serves to represent a particular product type such as 'BMW 325 i' or 'IBM Thinkpad T21'. To introduce a new product type, an object of `Product_Type` has to be instantiated first. Then one would assign the necessary number of feature types, either instances of `Standard_Feature_Type` or `Optional_Feature_Type`. Features may be available in various form (color, size, etc.). The range of available feature forms can be specified by assigning corresponding instances of `Feature_Value`. Associations between feature types (`replaces`, `requires`) allow one to express configuration constraints. All the requirements formulated in section 2 are fulfilled using this approach, whereby pricing mechanisms are abstracted. However, one must consider that only a part of all conceivable configuration constraints can be expressed (through the association between `Optional_Feature_Type` and `Feature_Type`). For instance, it would not be possible to specify that a sun roof is available only for black cars, because that would require one to refer to the state of a `Feature_Value`, not just to the existence of another feature. However, a limitation of this kind is not a serious drawback in view of the intended scope of application. It would not be economic anyway if a trading site reproduces the detailed configuration possibilities

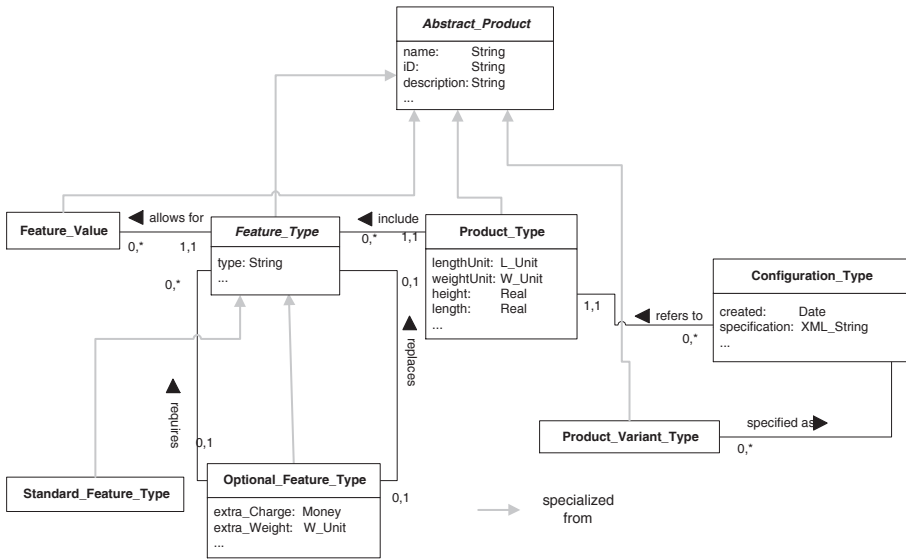


Fig. 2. Prototypical product model #1

of a complex product (a car, for example) in the same detail as the respective manufacturer. Therefore, for complex models the model is suited to describe subsets of all possible configurations. All individual configurations are managed as instances of **Configuration_Type**, whereby the attribute **specification** serves to describe an individual configuration in a suitable language (for example as an instance of a corresponding XML document type). Figure 3 clarifies this fact with an example. It represents standard and optional features of a particular sofa type and thereby expresses the set of possible configurations. In addition to that it shows an excerpt of a particular configuration that is represented as an XML string, which would be stored with an instance of the class **Configuration_Type**. Even when this approach supports a high degree of flexibility and concurrently a high level of integrity, it also has drawbacks. They are mainly related to the lack of support for re-usability. On the concept level, it is not possible to express that a product type is a specialization of an existing type. On the instance level, too, it may be helpful to re-use existing instances. Take, for example, certain makes of **Feature_Type**, like a particular car stereo make. While this is an important part of a product description, it could not be represented in an information system that is based on the model in fig. 2. The general feature type 'Car Stereo' would be represented by an instance of one of the subclasses of **Feature_Type**. From a conceptual viewpoint, a particular make would be an instance of this instance, which is not possible. As an alternative, one could use instances of **Feature_Type** to represent particular makes. Unfortunately, this approach would result in a large amount of conceptual redundancy.

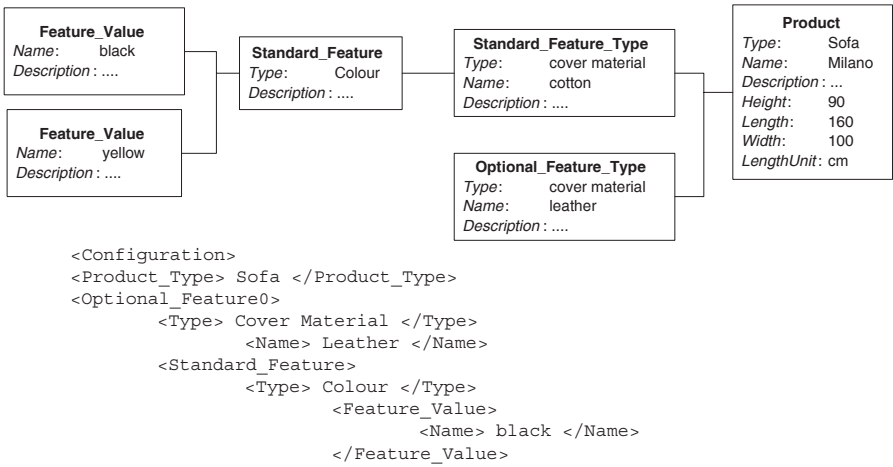


Fig. 3. Exemplary instantiation of the model with corresponding specification of a particular configuration using XML

4.2 Emphasis on Re-use

In cases where the shortcomings of the previous model are not acceptable, the representation of products should allow for expressing re-usability associations, both on a conceptual and an instance level. For this purpose, the object model in fig. 4 introduces additional concepts. They allow to represent product types as well as feature types on a higher level of abstraction. The model presented in fig. 4 requires to describe product types on a specific level, e.g. 'BMW 320i', 'BMW 325i' etc. This is also the case with feature types, e.g. the particular make of a car stereo. As outlined in 2, a more abstract representation would offer better chances for re-use. One would, for instance, introduce a product type 'Car' which could then be specialized into 'BMW 3 series' and further on into the concrete product type 'BMW 325i'. The class **Concrete.Product.Type** is intended to represent concrete product types. Similarly, one would represent feature types on a more abstract level, for instance 'Car Stereo' in general. A particular make of car stereo could then be instantiated from a corresponding type. The subclasses of the abstract class **Feature** serve to represent concrete features. Since many instances of **Concrete.Product.Type** may be assigned to an instance of **Product.Type**, particular features (instances of **Standard.Feature** and **Optional.Feature**) have to be assigned to an instance of **Concrete.Product.Type**. Unfortunately, the model would still allow to assign features to a product that are not assigned to feature types of the corresponding product type. Therefore additional constraints are required (see fig. 4). Notice that there is no need to link a feature to every feature type. Sometimes, the description of a feature type will be sufficient. Take, for instance, an air conditioning system in a car. To support conceptual re-use, the model includes a recursive association with **Product.Type** that indicates a specialization relationship. At the same time the

description of a concrete feature may require a more elaborate specification. For instance: the size of wheels. Since the relevant attributes of feature types may vary, there is need for an additional meta abstraction. It is provided with a meta attribute, i.e. an attribute that serves to specify a structure. The meta attribute `feature_Details` is specified by the class `Attribute.Spec`. It contains two tuple each of which contains the denominator of an object level attribute and the type of this attribute. The default value of this attribute contains the tuple `(Name, String)`, `(Description, String)` and `(Photo, String)`, which can be used to describe a particular feature (like an alloy wheel) on the object level. Since the structure of a feature type may vary, it is not possible to specify the structure of a particular feature in advance. Therefore a particular feature, stored as an instance of `Standard_Feature` or `Optional_Feature`, is specified as an XML document within the attribute `StrucDesc`. The structure of this document is defined with the corresponding instance of a subclass of `Feature.Type` as an XML document type definition (attribute `DTD` in fig. 4). Notice that an instance of `Standard_Feature_Type` or of `Optional_Feature_Type` may be associated with many instances of a corresponding subclass of `Feature` to express that there are options for specifying individual configurations. A product type consists of feature types.

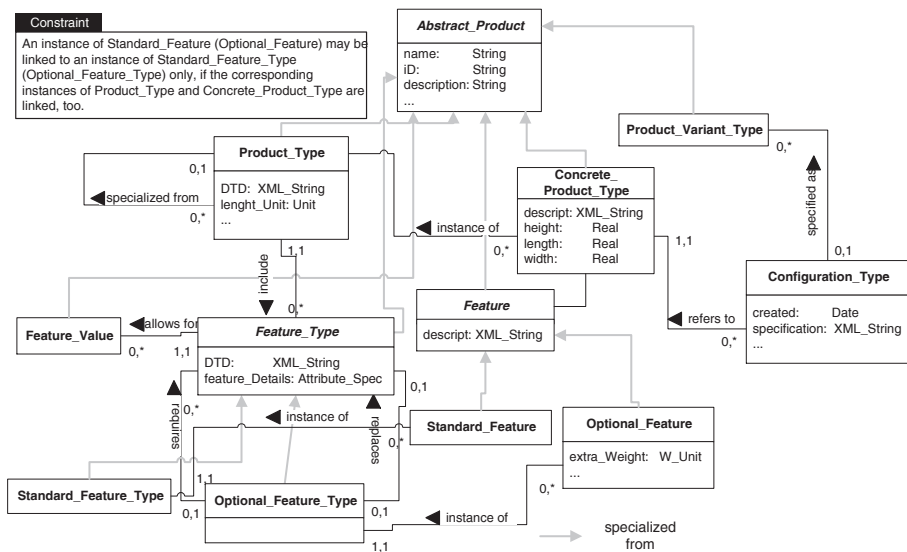


Fig. 4. Prototypical product model #2

With the specific structure of a feature type being defined in a corresponding DTD, the overall structure of a product type can be expressed as a DTD, too. It is composed of the DTDs of the associated feature types. A concrete product type, instance of `Concrete_Product_Type`, is then described by a number of

general attributes, like 'Width', 'Height' etc. and an XML string that contains a description of all features that are available with this concrete product type. Similar to the previous models, a particular configuration would be specified by an XML string that defines all selected features of a corresponding concrete product type. Notice that we do not allow for the "specialization" of feature types, because this could result in the notorious covariance problem ([6], pp. 621). This model fosters a higher level of re-use than the previous one. Nevertheless, it comes with two disadvantages. The first one is related to its complexity that is a threat to a system's integrity for two reasons. The implementation of additional constraints increases the chance of software bugs. At the same time, using it appropriately demands abstraction skills that may possibly not be expected of system users. While some of this complexity may be hidden from users, the second disadvantage is to accept only, if the similar product types share the same standard and optional features. Otherwise, it is not possible to take advantage of the specialization association. Cars, for instance, will usually not fulfill this condition: a feature that is standard for one product type may be an optional feature for another type of the same kind.

4.3 Compromising Flexibility for Re-usability: A Pragmatic Approach

The main problem we encountered with the previous model is caused by the variance of product types concerning possible configurations, i.e. their standard and optional features. The following model offers a pragmatic solution to the obvious conflict between flexibility (in terms of expressing configuration rules) and re-usability. Our analysis of products has shown that for many product types limited means to define configurations are sufficient. Therefore, the model compromises flexibility for re-usability. In contrast to the previous models, it does not distinguish between optional and standard feature types. Instead it includes only one `Feature.Type` (see fig. 5), which contains an attribute 'default' that allows for expressing whether a feature type is mandatory (like wheels). Additionally, it contains the meta attribute `Instance.Extra` that is specified by the class `Attribute.Spec` (see 4.2). It contains two tuple each of which serve to store the denominator of an object level attribute and the type of this attribute. For example, `(Extra, Boolean)` indicates that the object level attribute `Extra` is intended to store a boolean value that specifies whether the corresponding feature is optional or standard. To increase the level of re-use, an instance of `Feature` is not linked to an instance of `Concrete.Product.Type`. Instead, the instances of `Feature` to be used with a particular instance of `Concrete.Product.Type` have to be specified within the XML string (attribute `descript` in `Concrete.Product.Type`). Similar to the previous model, the structure of a product type as it can be deferred from the corresponding feature types is stored as an XML DTD with the attribute `DTD` of `Product.Type`. Also, the structure of the corresponding product type is stored as an instance of this DTD. This model allows for a high level of re-use both on the conceptual level and on the instance level. The first being fostered through

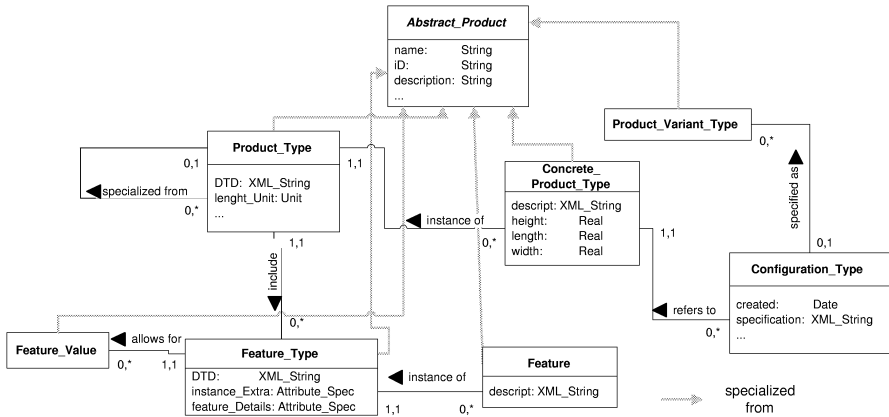


Fig. 5. Prototypical product model #3

the recursive 'specialization' association with `Product.Type`, the second by introducing meta attributes and corresponding XML strings. In comparison with the previous model, this model offers limited ways to describe configurations. While this may not be a serious disadvantage for many e-commerce platforms, the introduction of meta attributes and XML strings has its price: it certainly increases the complexity of a database that is instantiated from this model. Firstly, for a user to enter data, it is demanding to understand the various levels of abstraction. Secondly, for retrieving product types with certain features, using standard database search procedures is not sufficient. In addition to that, it is necessary to parse XML strings, too. While the first disadvantage can be compensated for to some degree by a user interface that hides most of the complexity, the second disadvantage can be reduced by using standard XML tools.

5 Conclusions and Future Work

In this paper we presented three prototypical conceptual models to represent products for e-commerce platforms. While all of them fulfill the requirements discussed in 2, each of them comes with specific strengths and drawbacks. Table 1 presents a comparison of the three models. 'Varying configuration rules' refers to a model's ability to handle varying configuration rules within the concrete product types that are "instances" of a product type the configuration rules were defined for.

The concepts for product modeling presented in this article were created during the development of a reference model for trading platforms on the Internet. The reference model exists currently as an UML object model with approximately 170 classes that is accompanied by numerous models of corresponding business

Table 1. Comparison of Models

	model #1	model #2	model #3
generalisation of product types	-	+	+
rich definition of configuration rules	+	+	o
re-use of feature types	-	-	o
re-use of features	-	+	+
varying configuration rules	-	+	+

processes. The representation of products within the reference model is similar to the last prototypical model. It covers, however, more aspects, like pricing, composite objects (to represent e.g. a computer system consisting of a PC, a screen and a printer) and packages. Our work on conceptual product modeling is part of the research project ECOMOD (‘Enterprise Modeling for E-Commerce’) that is funded by the German National Science Foundation. Apart from this application area we encountered similar problems with abstractions required for the design of knowledge management systems ([3]) as well as for the design of modeling tools. Therefore our future research is also aimed at investigating more general design patterns for dealing with multiple levels of abstraction.

References

- [1] ARIBA: *cXML User’s Guide. Vers. 1.1* Boston, London: Artech House (1998)
- [2] COMMERCE ONE: *cXML User’s Guide. Vers. 1.1* (2000) (www.cXML.org)
- [3] FRANK, U.; FRAUNHOLZ, B.; SCHAUER, H.: *A Multi Layer Architecture for Integrated Project Memory and Management Systems*. In: Khoshrow-Pour, M. (Ed.): *Managing Information Technology in a Global Economy: Proceedings of the 2001 IRMA International Conference*. Hershey, London, etc.: Idea Group Publishing (2001) 336–340
- [4] FRANK, U. *Delegation: An Important Concept for the Appropriate Design of Object Models*. In: *Journal of Object-Oriented Programming*. Journal of Object-Oriented Programming. Vol. 13, No. 3 (2000) 13–18
- [5] HÜMPEL, C.; RENNER, T.; SCHMITZ, V.: *BMECat Specification. Version 1.01*. Technical Report, Universität Essen (1999)
- [6] MEYER, B.: *Object-oriented Software Construction*. 2. Ed., Englewood Cliffs, NJ: Prentice Hall (1997)
- [7] OBI CONSORTIUM: *Open Buying on the Internet*. Draft v 0.6.5 (1999)
- [8] ROSETTANET: *RosettaNet Implementation Framework Specification*. Version 1.1 (1999)
- [9] SCHMITZ, V.; KELKAR, O.; PASTOORS, T.; RENNER, T.; HÜMPEL, C.: *Specification BMEcat: Version 1.2* Technical Report, Universität Essen (2001)

Seamless Personalization of E-commerce Applications

Juan Cappi¹, Gustavo Rossi¹, Andrés Fortier¹, and Daniel Schwabe²

¹ LIFIA-Facultad de Informática-UNLP, Argentina
{jcappi,gustavo,andres}@lifia.info.unlp.edu.ar

²Depto. de Informática, PUC-Rio, Brazil
schwabe@inf.puc-rio.br

Abstract. In this paper we present an original approach for personalizing complex Web applications, in particular e-commerce applications. This approach is based on a clear separation of concerns, namely: base application functionality, user profile management, and personalization rules, and supports seamless addition of personalization features (such as recommendations, special offers, individual interfaces, etc). We first explain our view of e-commerce applications as views on application models, and briefly explain why personalization functionality should be dealt by separating concerns. We next introduce a simple example and focus on different personalization patterns, emphasizing on behavior personalization. We show which design structures are the most appropriated for obtaining seamless extensions to existing software. We finally discuss some further aspects in building customized e-commerce software.

1 Introduction

Personalization has become a very important issue in e-commerce applications. The increasing popularity of the World Wide Web, the myriad of platforms supporting some kind of browsing, and the very nature of e-commerce have reshaped this problem. Not only we have to build applications customized to the individual; we must also cope with constant changes of personalization policies and rules (that in general follow the evolution of the business model itself).

Designing personalized e-commerce applications implies dealing with different concerns. It may mean building different interfaces (customized to a particular appliance), providing personalized navigation paths; offering different pricing policies, customized check-out procedures, etc. All these features involve managing information about the user (and his profile), implementing specific algorithms and interfaces, designing and applying rules and other related design problems.

User modeling and profile derivation from internet-gathered information has been extensively discussed in the literature [Perkowitz 00]. Recommendation algorithms and mechanisms (such as push vs. pull) have been recently surveyed [Schafer 99] and almost all electronic stores now include some kind of personalized behavior. A broader approach for understanding general requirements for customized Web Applications is presented in [Kappel 00].

However, while most of the “algorithmic” issues of personalization have been already studied, little attention has been paid to the modeling and design process of this kind of software; in particular, which design structures may help to cope with its increasing complexity. This aspect is critic not only because of the difficulties inherent to the problem but also because of the evolvable nature of (personalized) e-commerce applications. More precisely, as personalization requirements and policies change over time, software maintenance becomes a nightmare.

Suppose for example that product prices are personalized according to the buying history of the user; however, on a particular week there is a sale for certain products for which the mentioned policy does not apply. Or we decide to change the discount policy and apply it only when the user spent more than a certain amount of money. How do we guarantee software stability in this context? How do we simplify the addition of new rules? How do we support new features in the user profile that were not foreseen? How do we avoid melting business rules into the application code? The answer is separation of concerns.

By clearly understanding and decoupling the design concerns involved in a personalized e-commerce application, we can keep the software manageable; we can also provide a conceptual framework for reusing designs and design experience. We have a better way to understand the kind of interactions appearing in a personalized application from an abstract point of view (independent of the specific aspects of the application), and to simplify them by following existing design patterns.

We show that a clear separation of those concerns related with personalization allows adding personalized behavior to applications conceptual models with a minimum amount of code manipulation. We stress that reasoning over design objects gives us better insight on the personalization process.

The structure of this paper is as follows: We first introduce the OOHDM approach for building e-commerce applications. Next, we discuss the problem of designing personalized applications. We identify the most common personalization patterns and analyze the particular design concerns they involve. We introduce a stereotypical example to illustrate our ideas briefly and show personalization hot spots in the example. We next introduce personalized wrappers and rule objects; for each design mechanism we discuss the problem which motivates it and its solution. We finally explain how to scale-up towards a generic architecture for personalized applications and discuss some ongoing research issues.

2 Our View of E-commerce Applications

The key concept in OOHDM is that Web (in particular e-commerce) application models involve a Conceptual, a Navigational Model and an Interface Model [Schwabe 98]. Those models are built using object-oriented primitives with a syntax close to UML.

The concern of the conceptual model is to represent domain objects, relationships and the intended applications’ functionality. As we show later in this paper, most personalization mechanisms involve dealing with objects and algorithms that are expressed as part of the conceptual model.

The cornerstone of the OOHDM approach is that the user does not navigate through conceptual objects, but through navigation objects (nodes). Nodes are defined as views on conceptual objects, using a language that is similar to object database view-definition approaches. As we consider Web applications as hypermedia applications, we define links connecting nodes, as views on conceptual relationships. Nodes are further organized in sets called navigational contexts. Indexes provide shortcuts for accessing nodes. The complete syntax for nodes, links, contexts and indexes definition can be found in [Schwabe 98]. OOHDM provides a straightforward way to build applications customized to different user roles building different navigational models (views) for each profile.

Finally, the abstract interface model specifies the look and feel of navigation objects together with the interaction metaphor. Separating the interface from the navigation specification allows us to cope with varying interface technologies in a modular way. As interface objects are specified as Observers [Gamma 95] of navigation objects, we also have a simple way for customizing the interface to different Web appliances (as browsers, cellular phones, etc) [Schwabe 98].

3 Modeling and Design of Personalized Applications

We have mined recurrent personalization patterns in Web applications [Rossi 01a]. These (coarse grained) patterns allow us to focus on *what* can be personalized before addressing which concerns are involved in the personalization design process. Following the OOHDM framework (that partitions the design space into conceptual, navigational and interface models) we can personalize:

- the algorithms and processes described in the conceptual model.
- the contents and structure of nodes and the link topology in the navigational model
- the interface objects and their perceivable aspects and the interaction styles.

These personalization aspects can be addressed at the role or at the individual levels. As OOHDM naturally supports role-based personalization we do not discuss it further in this paper. For the sake of conciseness we will neither discuss interface personalization though the basic principles discussed later can be easily apply to this aspect.

Once we understand the different personalization patterns in an abstract way, we must discuss how to relate specific applications' requirements with these patterns and then map these patterns onto concrete designs. For example, if we need to personalize products' prices in an e-store, we first realize that this is a particular case of node content personalization and then we use the most appropriate abstractions for materializing design. In this paper we present some general design solutions to these problems; we present them in a domain and application independent way, so they can be reused in different systems and modeling approaches.

There are basically three concerns related with personalization: the user profile, the personalization rules and the application of those rules for a particular individual. Even in trivial e-commerce applications it can be shown that hard-coding some of these aspects in core application objects may yield a difficult to maintain application

as shown in section 4. To make matter worse, we usually need to improve existing applications adding personalization features that were not foreseen (see for example the evolution of amazon.com, cnn.com or yahoo.com); we should do it without affecting existing application logic.

The key for obtaining a good design is recognizing that the pace of evolution of customization rules (and the associated user profiles) is faster than changes in the basic application model so they must be clearly decoupled. We next introduce an example to discuss these issues more concretely.

4 An Example

Suppose an electronic store that sells *products* to *customers*. The customer may put products in his *shopping cart*, indicating the number of items of each product. When he decides to buy, the *check-out* process generates an *order* containing the products, *paying mechanism*, *shipping address* and *delivering options*; he choose if the product should be gift-wrapped. Each customer has an *account* containing his buying history. In Figure 1 we present an object model representing key features as classes.

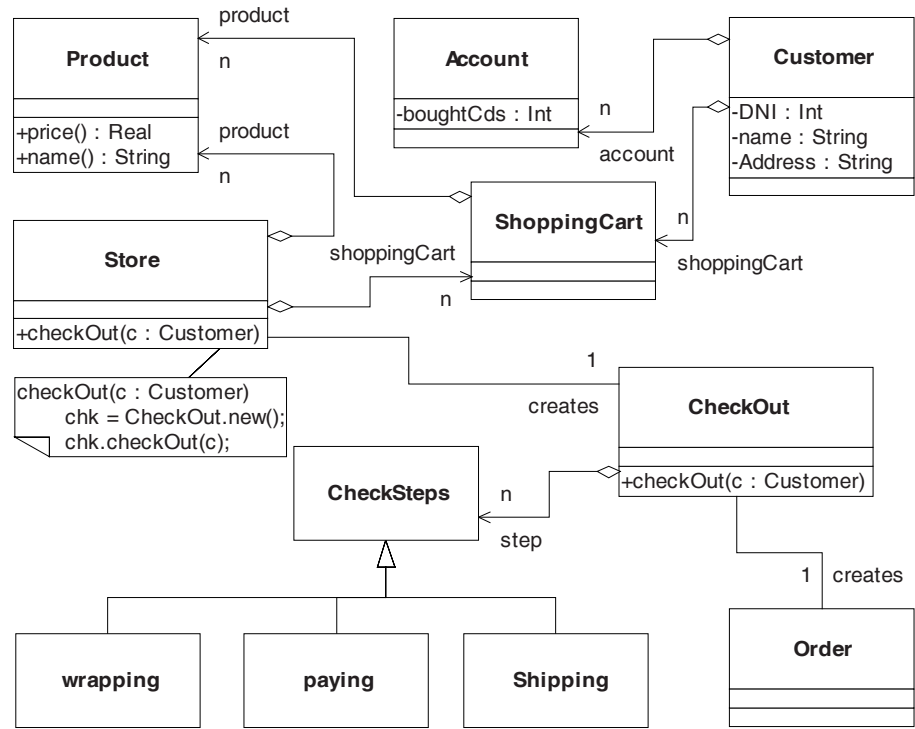


Fig. 1. Conceptual Model for the e-store.

While variations in the domain model such as adding new products or paying mechanisms can be solved by sub-classing and composing objects, if we want to

introduce some personalization capabilities, the problem gets worse. For example suppose that we want to personalize product prices; one possibility is to let products delegate the price computation to the corresponding customer object as shown in Figure 2. The same strategy can be used with other object's attributes.

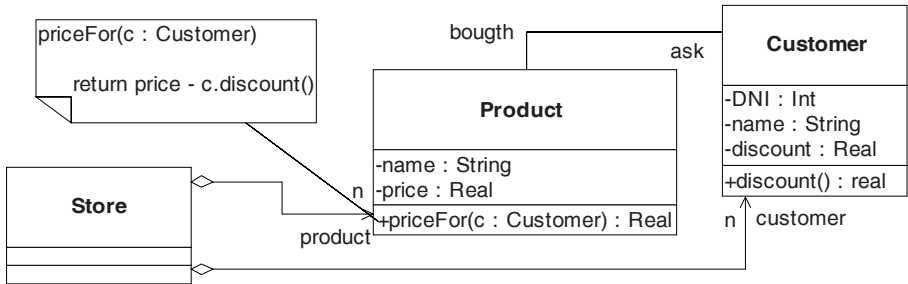


Fig. 2. Hard-coding Personalized Prices in products

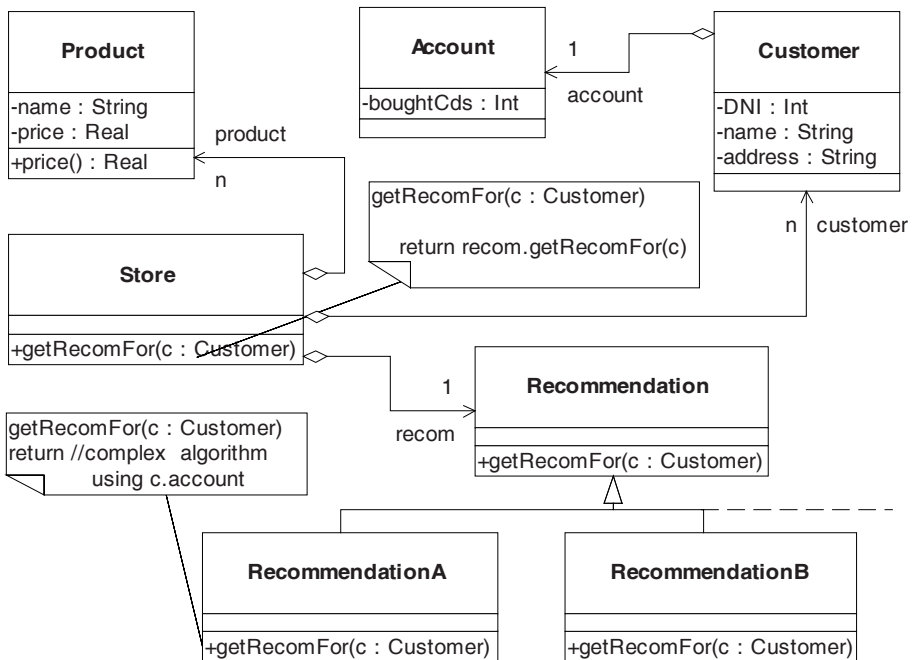


Fig. 3. Adding recommendations to the base model

The problem with this approach is that it works well if no new customization rule must be added or if we don't need different rules in different use contexts, e.g.: when the user browses the product, he gets one price (his price) but when he checks-out, the price may be personalized differently (for example taken into account a special offer when buying some combination of products). If this happens we will end re-designing core business classes constantly and sooner or later the code will become messy.

Under those conditions, it is interesting to notice that recommendations, in fact the most popular kind of personalization feature, are very easy to engineer as shown in Figure 3, where algorithms are implemented using the Strategy design pattern [Gamma 95]. Instead of implementing the recommendation as a method in class Store, different recommendation algorithms are organized in a class hierarchy and thus we can switch algorithms dynamically. Recommendation algorithms will thus interact with customer objects and their accounts to obtain the desired results.

However, if we have different algorithms according to the user profile, the solution in Figure 3 does not scale-up as it may involve a complex if clause for associating the algorithm to the profile. The problem is that we are coupling the basic business behavior (e.g. in class Store) with one that depends on the user profile.

In the following section we elaborate this discussion and show a set of design micro-architectures for coping with personalization complexity.

5 Separating Personalization Concerns

The key strategy for obtaining evolvable personalized e-commerce applications is decoupling the main concerns of these applications. From the above discussion we conclude that the core application logic should be separated from personalization rules. Other authors have also suggested keeping business rules separated from business objects as the former tend to vary faster than the latter [Arsanjani 99, Yoder 01].

As said before, the personalization logic itself involves three different aspects: the rules, the user profile and the application of those rules to particular business objects. Some of these concerns may partially overlap with the core objects as shown in the example. Even though the application was not conceived as personalized, it contains objects that will be useful as part of the customer profile (as Customer itself and its account), and behaviors for keeping the user profile up to date (such as those adding products to the customer's account). We next show step by step our approach for adding personalized behaviors in a non-intrusive way.

5.1 Personalizer Wrappers

The first problem to solve deals with changing the behaviors of some core business objects to personalize them. A simple instance of this problem is the method returning the price of a product. A more elaborated example can be found if we want to personalize certain steps of the checkout procedure. There are two constraints in this problem: we don't want to mix the basic code in the product with the "new" code for personalization and we also want to be able to easily "switch" these personalized behaviors in different contexts, e.g. prices when you buy one product, many products, in a special offer, etc.

The best solution to this design problem is based on the Decorator pattern [Gamma 95], which is used to dynamically add new behaviors to a given object. For each class we want to personalize we can define a corresponding wrapper and connect it to the personalized object in an instance basis (i.e. some objects may not be decorated). The

solution is shown in Figure 4, where we can see (in 4.a) the concept of wrapping. It consists in intercepting messages, eventually processing them without intervention of the base object and returning the result. Wrappers allow changing the behavior of an object without re-writing its class; as wrappers work at the instance level, we can wrap only some objects in a class. Figure 4.b shows the corresponding UML diagram.

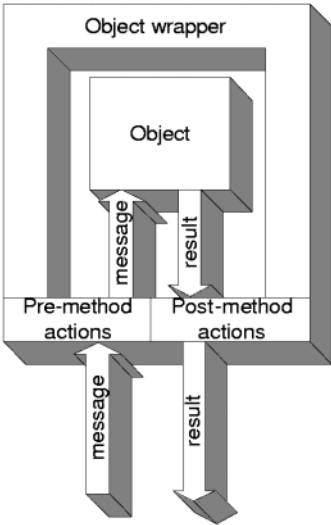


Figure 4.a: Functionality of a Wrapper

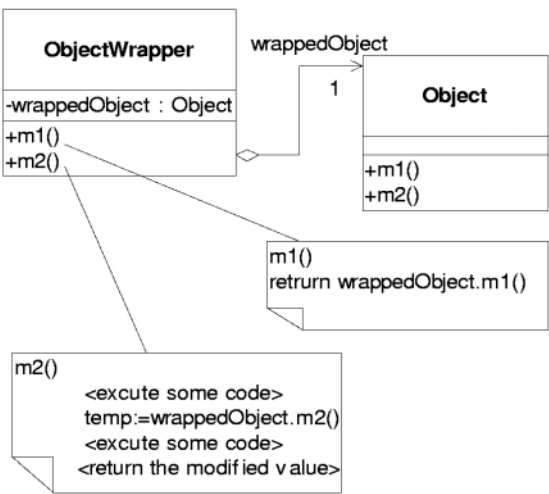


Figure 4.b: UML design for wrappers

Fig. 4. Decorators for personalized behavior

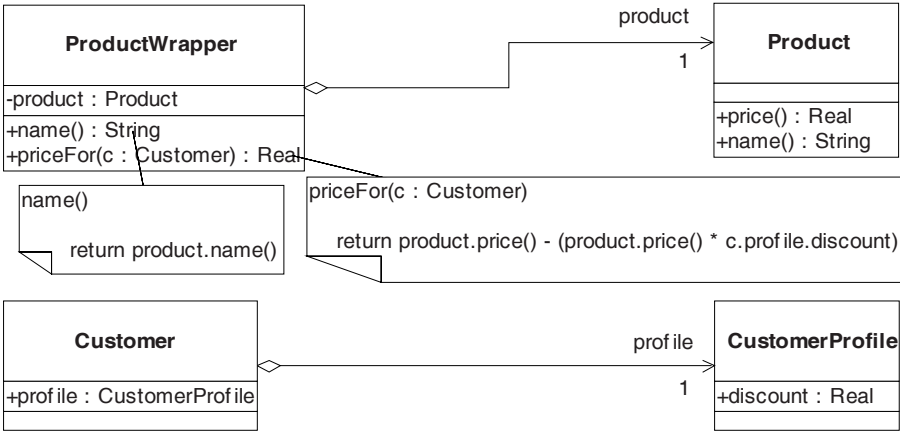


Fig. 5. Implementing personalized behaviors in wrappers

A first approach would be to allocate the personalized behavior in the wrapper, which might end up collaborating with other objects to complete the task. Following with the example of the product's price, the wrapper traps the message *price*, asks the user

profile the discount that should be applied and calculates the real price (Figure 5). Those behaviors that are not personalized are simply delegated to the original object. This same idea can be used for personalizing the checkout process or some of its steps (delivery and paying options, etc).

Even though this solution seems to work fine, we might think on applying a different policy to each particular customer. Of course, it is completely unacceptable to write a wrapper for each aspect for each customer (think of personalizing three aspects of an object, which can be handled in four different ways each: you end up writing 64 wrappers!). Other approach would be to write a potentially huge case statement to decide which algorithm to apply; there is no need to say that this idea won't scale up and the code ends completely messed up. An elegant solution to this problem can be addressed by using the Strategy pattern [Gamma 95], which decouples the algorithms from their implementation. Using this approach, the customer profile delegates the behavior for obtaining the discount to an other object; the wrapper would be just in charge of triggering its execution after getting the corresponding algorithm as shown in Figure 6. Notice that we just delegated the task of selecting which algorithm to apply to the Profile object. Now, we are able to choose different algorithms depending on the customer, and as a side effect we can even change those algorithms in run time as the user profile evolves.

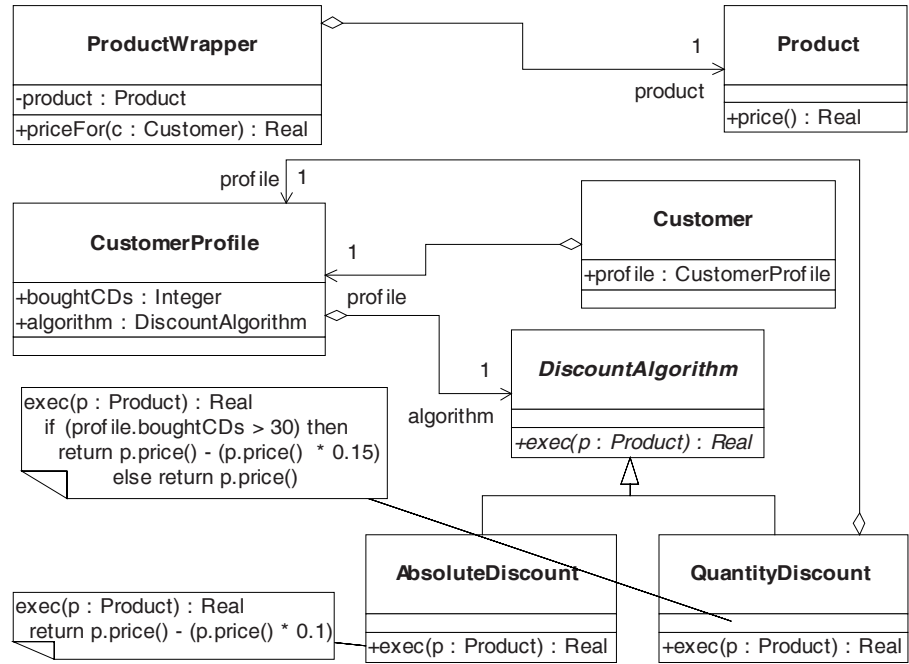


Fig. 6. Decoupling algorithm implementation

Since the way in which we personalize an aspect of an object is generally more than just an algorithm, we may call the wrappers ‘personalizers’, i.e. objects that take care of personalizing a particular aspect. Summarizing, the main idea is that we engineer

each personalized aspect separately and configure the wrapper as necessary, maybe in a per-instance basis.

To show how this approach helps us to cope with more complex situations, suppose that we need to personalize different aspects of the same object for different users; for example while one user has a personalized price, another has a personalized delivery mechanism. This is the typical case when users can select what can be personalized such as in *my.yahoo.com*. The solution is to delegate the responsibility of handling the appropriate personalizer to the corresponding user profile object, as shown in Figure 7 (where we personalize price and comments on a product). In this solution we further decouple the process of intercepting the message from the personalization code.

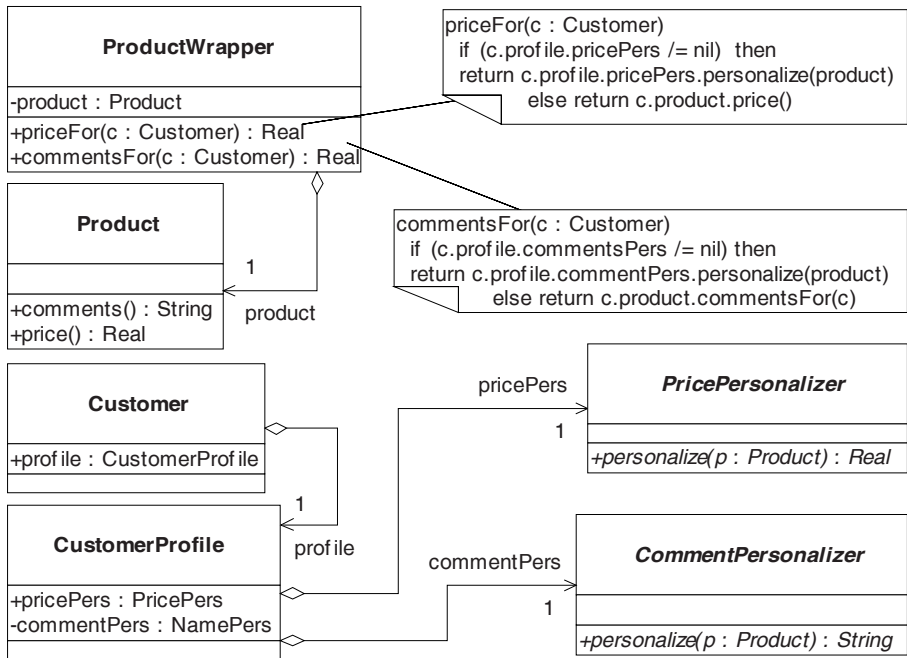


Fig. 7. Personalizers that depend upon the user profile

The product wrapper (See method: *priceFor*) in Figure 7, checks if the profile contains the corresponding personalizer; if it does not exist it just delegates the behavior to the corresponding product object.

This last design can be improved by using the Null Object pattern [Woolf 96], which helps us eliminate the if statement in the Product Wrapper. In Figure 8 we show how we define a hierarchy of personalizers. While *PricePersonalizerA* and *PricePersonalizerB* implement different price policies, the *NullPricePersonalizer* just delegate the task to the original Product object. Null objects are an elegant way to obtain uniform code because they help to eliminate complex if clauses.

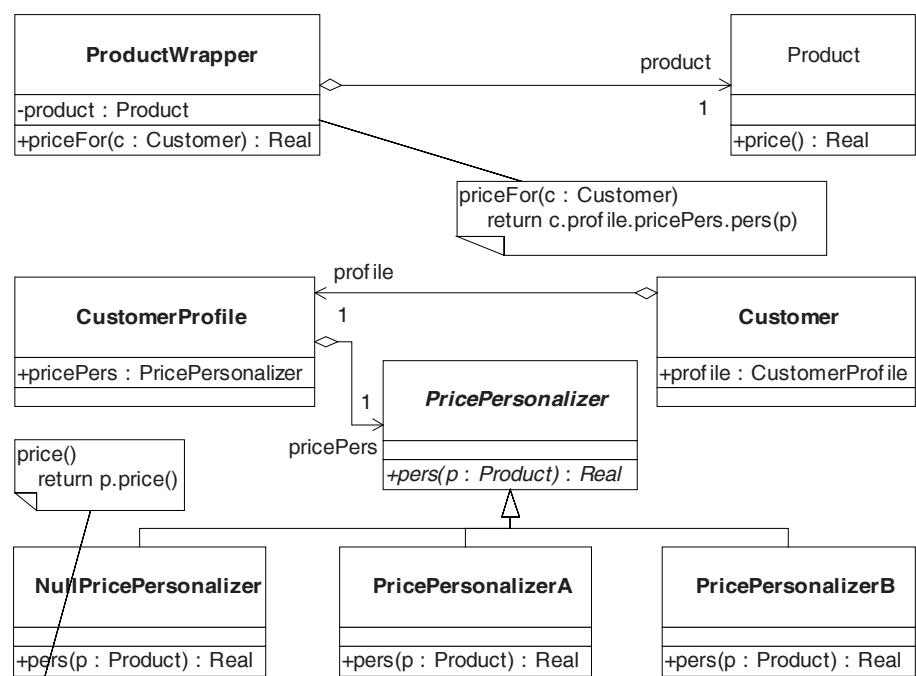


Fig. 8. Using the Null Object pattern

Personalizers allow decoupling the core business model from the personalized behaviors. However when customization rules are complex, we need to further decouple the rules from these objects as shown in the next sub-section.

5.2 Rule Objects

Personalization (or more generally business) rules are usually expressed in logical terms in the following way: *if a predicate <p> is satisfied, the actions <a₁,a₂,...,a_n> should be executed*. As rules are not first-class objects we need to analyze the best way to map them into design constructs.

As said before, personalizers have the responsibility of executing the personalization code: in fact they implement the concept of a rule.

This solution has two main problems. First, as rules may involve many different conditions and corresponding actions, personalizers may become monolithic, as they will contain complex *if then else* clauses. The best solution is to decouple Conditions and Actions from the Rule as shown in Figure 9. Moreover, when mapping this design to modern object-oriented languages (such as Java or Smalltalk), conditions and actions can be implemented as light-weighted objects such as blocks (or inner classes in Java) thus simplifying the design shown in Figure 9.

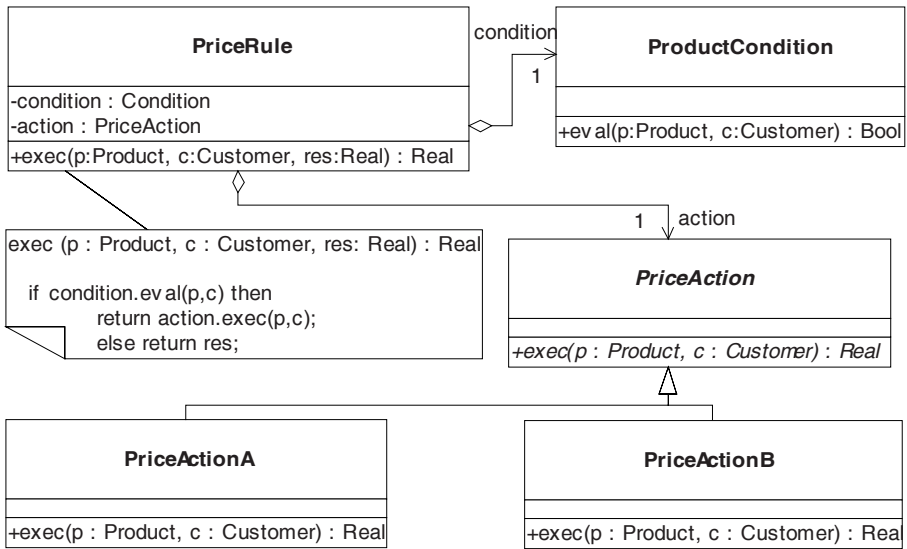


Fig. 9. The design of a rule

Using Conditions and Actions as shown in Figure 9, we can configure different price rules by combining different instances of PriceAction and ProductCondition. The reader should notice that this approach is easy to generalize by defining abstract classes for Conditions and Actions.

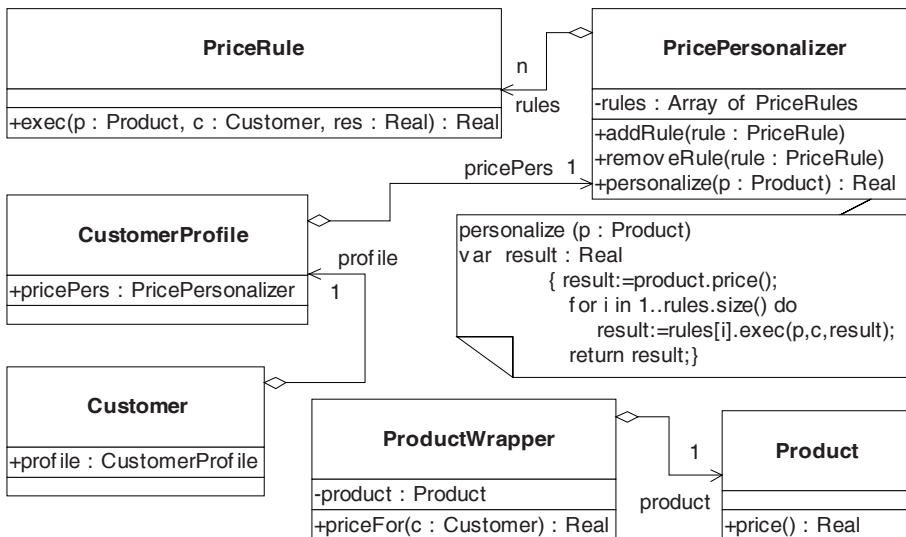


Fig. 10. Decoupling rules from personalizers

The second problem we should solve is related with the evolution of rules. As previously discussed, business rules tend to change quickly; new rules related with an aspect may be added or eliminated. In the case of pricing policies we may have rules that apply when the customer bought many products, others related with the current order, etc. The solution is to design rules as separate artifacts (thus, decoupling them from personalizers) in order to simplify their organization. This solution, shown in Figure 10, treats rules as objects that may be triggered from personalizers or eventually from other objects as explained in Section 5.4.

Notice that personalizers act as Facades [Gamma 95] and they may eventually implement a policy for handling rules, for example when the same aspect may be personalized using different rules. We can also reuse whole rules when they may be applied to different objects (e.g. product sub-classes, or interfaces), or even define hierarchies of personalizers that may be applied to different products.

5.3 Dealing with Evolving User Profiles

The user profile is an important component in every personalized application. As shown before, the user profile may not only contain plain information about the user such as his shopping history (see Figure 1) but may also contain rules, references to algorithms, etc. In our design approach the user profile should be clearly separated from business objects and should grow in a transparent way. However this is not always possible: many information stored in the user profile is generated by core business operations (e.g.: buying a product). Personalization rules may even require this information to evolve: suppose for example that we want to apply some price reduction to those customers who add some comments about products (for example after reading a book). In this case, the evolution of the user profile implies an interaction that will be handled (at least at first) by business objects and will require adding some methods and even changing the structure of the profile. From the designer point of view, however, it is important to characterize operations that affect the profile, i.e. those that add information, those that associate personalization rules with it and those that allow rules to query information about the user. These operations should be kept isolated and be decoupled from the rest of the business model when possible.

5.4 Putting All Pieces Together

In the previous paragraphs we have discussed the most important design problems a developer has to face when personalizing e-commerce applications. Each critical concern has been consciously decoupled yielding a modular architecture that can evolve seamlessly together with the business model's evolution. Each component (i.e.: wrappers, rules and the user profile) can be engineered separately provided that they conform to common protocols. The ideas underlying this architecture can be used as they are, or just as guidelines (or patterns) during e-commerce application building. For example, it is possible to get simpler designs when the forces that drive the application's evolution do not require that all components are completely decoupled (e.g. writing the rules code in personalizers).

It is interesting to remark that these micro-architectural constructs can be used to customize e-commerce applications, even when the customization conditions do not

involve individual users. For example suppose that we need to implement a marketing strategy that reduces products' prices during Christmas season. We can also think about wrapping products that intercept this behavior and delegate it to a set of rule objects that check the environmental conditions (e.g. date) and decide the corresponding action.

All these components can be materialized in abstract classes that can be eventually sub-classified for each particular application. The process of personalizing an e-commerce application will then consist in creating instances of corresponding wrappers, personalizers and rules and "plugging" them into base classes as shown earlier in this paper.

6 Conclusions and Further Work

In this paper we have presented a design approach for building personalized e-commerce applications. This approach is based on a clear separation of concerns, namely base business logic, user profile, personalization rules and the application of rules.

We have shown that personalization issues should be carefully considered when building the application's model. In particular we claim that design components related with personalization should not be hard coded in core application objects, but rather built as separated artifacts and conveniently composed with the objects they personalize. We showed that applying well-known patterns (such as Decorator, Strategy and Facade) we can seamlessly extend e-commerce applications with personalization features. The approach is non-intrusive because it provides architectural mechanisms for adding extra functionality (e.g. personalization) without re-coding the base business logic.

We are now developing visual tools for simplifying the personalization process by allowing the designer to plug personalizers to base objects easily. This approach has been used successfully for letting designers describe rules visually [Yoder 01].

We are finally exploring how to extend our approach to a more general scope of customization, not limited to individual customization. The whole architecture can be reused, though rule managers should provide additional functionality for other customization rules. In this case, wrappers should also provide the context of customization (thus allowing rules to "know" which objects may contain information useful for the process). Our approach can then be applied to more general frameworks like the one in [Kappel 00] to solve the problem of ubiquity and customization in e-commerce.

References

- [Arsanjani 99] A. Arsanjani. "Analysis, Design, and Implementation of Distributed Java Business Frameworks Using Domain Patterns" in Proceedings of Technology of Object-oriented Languages and Systems 30, *IEEE Computer Society Press* 1999, pp. 490-500
- [Gamma 95] E. Gamma, R. Helm, R. Johnson, J. Vlissides: "Design Patterns. Elements of reusable object-oriented software", Addison Wesley 1995.

- [Kappel 00] G. Kappel, W. Retschitzegger and W. Schwinger. "Modeling Customizable Web Applications - A Requirement's Perspective". In Proc. International Conference on Digital Libraries: Research and Practice, Kyoto 2000.
- [Perkowitz 00] M. Perkowitz, O. Etzioni: "Adaptive Web Sites" In Comm ACM, August 2000, p.p. 152-158.
- [Rossi 01a] G. Rossi, D. Schwabe, J. Danculovic, L. Miaton: "Patterns for Personalized Web Applications", Proceedings of EuroPLOP 01, Germany, July 2001.
- [Schwabe 98] D. Schwabe, G. Rossi: "An object-oriented approach to web-based application design". Theory and Practice of Object Systems (TAPOS), Special Issue on the Internet, v. 4#4, pp.207-225, October, 1998.
- [Schafer 99] Schafer, J. B.; Konstan, J.; Riedl, J.; "Recommender Systems in E-Commerce", Proc. of E-Commerce'99, Denver, USA, ACM, 1999, pp.158-166.
- [Woolf 96] Bobby Woolf: "The null object pattern". Proceedings of PloP'96, Pattern Languages of Program Design, 1996.
- [Yoder 01] J. Yoder, F. Balaguer, R. Johnson: Adaptive Object-Models: With Application to Medical Observations. To be published in Proceedings of OOPSLA 2001.

Discovery of User Preference through Genetic Algorithm and Bayesian Categorization for Recommendation

SuJeong Ko and JungHyun Lee

Department of Computer Science & Engineering
Inha University

Yong_hyen dong , Namgu, Incheon, Korea
{sujung@nlsun.inha.ac.kr}, {jhlee@inha.ac.kr}

Abstract. Recent recommender system uses two complementary techniques. Collaborative filtering uses a database about user preferences to predict additional topics. Content based systems provide recommendations by matching user interests with topic attributes. In this paper, we describe a method for discovery of user preference by using hybrid two techniques for recommendation that allow the application of machine learning algorithm. The method generates recommendations based on clustering user and categorizing items with feature selection through association word mining by Apriori algorithm. We use Genetic algorithm to group users based on items categorized by Naïve Bayes classifier. Then, we recommend web documents to user based on grouped user preference and information of categorized items. We evaluate our method on a large database of user ratings for web document and it significantly outperforms previous proposed methods.

1 Introduction

Current approaches to Web personalization generally fall into three major categories: manual decision rule systems, collaborative filtering systems, and content based filtering agents[10,18]. These are several well-known drawbacks to these traditional content based or rule based filtering techniques for personalization. The profiles in these techniques are often static, obtained through user registration, and thus the system performance degrades over time as the profiles age. Content based systems provide recommendations by matching user interests with item attributes. Sometimes, there are a large number of item attributes, and existing systems rely heavily on preprocessing steps that select or extract important features from the items[3,6,7]. Collaborative filtering has tried to address some of these issues, and, in fact, it is the predominant commercial approach in most successful e-commerce systems[5]. These techniques generally involve matching in real time the ratings of a current user for objects with those of similar users in order to produce recommendations on other objects not yet rated by the user. However, collaborative filtering techniques have their own potentially serious limitations. For instance, it becomes hard to scale

collaborative filtering techniques to a large number of items, while maintaining reasonable prediction performance and accuracy. Part of this is due to the increasing sparsity in the ratings data as the number of items increases, as well as due to the increasing computational cost of determining user to user correlation in real time for a large number of items and users[21]. For the purpose of solving these problems, model based method are proposed. It is usually superior to memory based methods in this respect, as they can impose constraints via the models[4,8,17]. Difficulties also arise when a new product comes into the market and thus has no previous preference information-first rater problem. Integration of content-based and collaborative approaches is a promising paradigm to alleviate this problem[22].

In this paper, we describe a method for discovery of user preference for recommendation that allows the application of machine learning algorithm. The method generates recommendations based on clustering user and categorizing items with feature selection through association word mining. It categorizes items using Naïve Bayes algorithm in order to reduce the dimensionality and sparseness of the score matrix. We use Apriori algorithm in order to select feature of item. Items using in this paper are web document in term of computer science. Then, proposed method clusters user using Genetic algorithm[14] based on categorized items in order to reduce noise and sparseness of the score matrix resulted from many users.

We evaluate our method on a large database of user ratings for web document and it significantly outperforms previous proposed methods.

2 System Overview

Fig. 1 shows system overview for discovery of user preference through Bayesian categorization and Genetic algorithm. Fig. 1 is composed of content based filtering and collaborative filtering for recommendation.

In content based filtering, item is categorized into class based on content of item. Items using in this paper is a web document in term of computer science that is gathered by HTTP down loader. Consider a dataset with m "items" having a feature vector of length p , and n "users". Let $U=\{u_i\}(i=1,2,\dots,n)$ be the set of users and $I=\{d_j\}(j=1,2,\dots,m)$ be the set of items. Let $R=\{r_{ij}\}(i=1,2,\dots,n \text{ and } j=1,2,\dots,m)$ be the matrix of user-item in which entry r_{ij} denotes user u_i 's preference rating for item d_j . Naïve Bayes classifier categorizes item d_j in matrix R into class.

In collaborative filtering, we group users into cluster based on items classified by Naïve Bayes classifier. Previously, $R=\{r_{ij}\}(i=1,2,\dots,n \text{ and } j=1,2,\dots,m)$ was the matrix of user-item. Let $C=\{class_k\}(k=1,2,\dots,N)$ be the set of categorized items. We produce $R_I=\{c_{ki}\}(k=1,2,\dots,N \text{ and } i=1,2,\dots,n)$ through transforming matrix R . Matrix R_I is the matrix of class-user in which the entry c_{ki} denotes to be the number of items categorized into $class_k$ among items rated by user u_i . This paper only uses interesting items to an individual user among items rated by user in order to discover user preference. The preference ratings in this paper are discrete into 6 levels, as 0,0.2,0.4,0.6,0.8, and 1. We define an item as interesting item to an individual user if his preference rating for this item is greater than 0.5. Genetic algorithm groups people

in collaborative filtering database into cluster. New user is grouped into cluster with the highest similarity among clusters in collaborative filtering database. User preference is discovered by users' preference in cluster where new user is grouped.

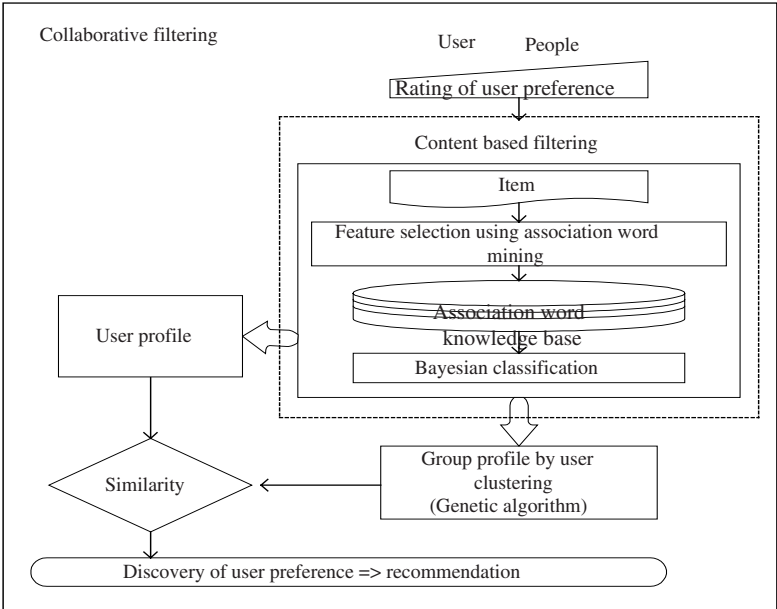


Fig. 1. System overview for discovering user preference

3 Content Based Filtering

Content based filtering categorizes an item into class based on content of an item. Feature selection method for classification is association word mining[12]. Bayesian classification method is a Naïve Bayes classifier based on association word knowledge base[11].

3.1 Feature Selection

We adopt the commonly used ‘bag-of-words’[16] document representation scheme(vector space model), in which we ignore the structure of a document and the order of words in the document[15]. ‘bag-of-words’ is composed of nouns pruned stop-list from results after morphological analysis[19,20]. In this paper, we represent ‘bag-of-words’ as ‘bag-of-association words’. The feature vectors represent the association words observed in the documents. The association word-list in the training set consists of all the distinct words that appear in the training samples after removing

the stop-words. The AW (association word)_list is defined to be: $AW = \{(W1r: w_{11} \& w_{12} \dots \& w_{1(r-1)} => w_{1r}), (W2r: w_{21} \& w_{22} \dots \& w_{2(r-1)} => w_{2r}), \dots, (Wkr: w_{k1} \& w_{k2} \dots \& w_{k(r-1)} => w_{kr}), \dots, (Wpr: w_{p1} \& w_{p2} \dots \& w_{p(r-1)} => w_{pr})\}$. Here, each of $\{w_{k1}, w_{k2}, \dots, w_{k(r-1)}, w_{kr}\}$ in $(w_{k1} \& w_{k2} \dots \& w_{k(r-1)} => w_{kr})$ represents a word for composing association word. “p” in AW represents the number of association words in a document. “r” in AW represents the number of words in an association word. “&” at pairs of words means that pairs of words have a high degree of semantic relatedness. “ $w_{k1} \& w_{k2} \dots \& w_{k(r-1)}$ ” is antecedent of association word $(w_{k1} \& w_{k2} \dots \& w_{k(r-1)} => w_{kr})$ and “ w_{kr} ” is consequent of association word $(w_{k1} \& w_{k2} \dots \& w_{k(r-1)} => w_{kr})$. Each of $\{W1r, W2r, \dots, Wkr, \dots, Wpr\}$ represents association word.

Apriori algorithm[1,2] extracts association rule between words through data mining. Procedure for mining association rule between words consists of two stages. In the first stage, composition having transaction support in excess of min_support is found to constitute frequent word item. In the second stage, frequent word item is used to create association rule from database. As for all frequent word item(L), find subset instead of all empty set of frequent word item. As for each subset(A), if ratio of support(L) against support(A) is not less than min_confidence, rule of A->(L-A) type is displayed. Support of this rule is support(L). In order to extract the most proper association word, confidence should be fixed at not less than 85 and support should be not more than 22[12].

According to this method of feature selection, we can describe item $\{d_j\}$ to be as following.

$$\{d_j\} = \{W1r, W2r, \dots, Wkr, \dots, Wpr\}, \quad j=1, 2, \dots, m$$

3.2 Categorizing Item Using Naïve Bayes Classifier

Naïve Bayes classifier can classify documents through learning phase and classification phase[23]. The learning phase gives a probability to association word mined from training document by Apriori algorithm. Equation (1) is used to give a probability to association word Wkr within classID. In this paper, a probability of association word Wkr within classID is expressed as $P(Wkr | classID)$. In that case, n is the total number of mined association word and n_k is the number of association word that corresponds with association word Wkr in class. classID is the label of class, and |AW| is the total number of association words within all classes.

$$P(Wkr | classID) = \frac{n_k + 1}{n + |AW|} \quad (1)$$

Learning process is divided into the phase for accumulation and the phase for granting probability. In the phase for accumulation, the number of times is accumulated if association words within documents exist within a class. In phase for granting probability, outcome of accumulation is applied to Equation (1) and probability is granted to association word. Through this process, probability is added to association word in class.

Naïve Bayes classifier classifies test document using Equation (2). To do this, Apriori algorithm extracts characteristic in the form of association word from test document as described in Section 3.1. We define characteristic of test document(D) to be $\{dWlr, ..., dWkr, ...\}$. “d” of $dWkr$ indicates association word extracted from test document.

$$class = \arg \max_{classID=1}^N P(classID) \sum_{k=1}^m P(dWkr|classID) \quad (2)$$

In Equation (2), class where document(D) is to be classified is $class$ and the total number of classes is N . $P(dWkr|classID)$ is the probability according to Equation (1) and $P(classID)$ is the probability that it is classified as $classID$.

4 Collaborative Filtering

Previously, we defined an item “interesting” to an individual user if his preference rating for this item is greater than 0.5. To do this, we denote R_i to be as following. Table 1 shows an example of matrix R_i with 18 users and 19 classes.

$$R_i = \{d_j \in class_k, 0 < j \in m, 0 < class_k \in N \mid c_{ki} = c_{ki} + 1, R_{ij} > 0.5\}$$

Table 1. An example of matrix R_i

class	Users																	
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}	u_{17}	u_{18}
1	3	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0
2	0	0	2	0	0	0	0	0	0	0	0	2	0	0	1	0	0	1
3	2	0	4	0	1	1	1	2	0	2	0	0	0	0	2	2	1	3
4	2	2	5	4	2	2	3	1	0	1	0	0	0	0	1	1	2	1
5	1	0	1	0	2	0	1	3	0	3	0	0	0	0	3	2	1	2
6	2	3	1	5	3	3	2	1	0	3	0	0	0	0	1	4	3	1
7	3	0	0	0	1	1	1	2	0	0	0	0	0	0	0	0	0	0
8	1	4	1	1	1	1	1	4	0	1	0	0	0	0	1	1	1	1
9	1	1	2	1	2	0	2	1	0	1	0	1	3	3	3	2	2	2
10	1	0	1	0	1	2	0	1	1	0	0	0	0	0	0	0	1	0

4.1 Optimal User Clustering

Genetic algorithm groups users into cluster. Genetic algorithm groups users by using gene, chromosome, and population. Gene indicates entry c_{ki} in matrix R_i of class-user. In this paper, we express gene as a real number. The number of genes is the number of users. Chromosome expresses the row in matrix R_i . Therefore, the number of chromosome is the number of classes. Population is the whole entries in matrix R_i . Population evolves through initialization, fitness calculation, recombination, selection,

crossover, mutation, and evaluation. Evolved population is assessed and it is decided whether to continue its evolution to next generation.

In initialization stage, user is defined as gene. Gene in Table 1 is changed to real number because we use gene as a real number. In addition, for the purpose of reducing the complexity of computing, we change gene to small real number by Equation (3). Then, we transform entry c_{ki} in R_i into entry c'_{ki} in R'_i using Equation (3). Matrix R_i is transformed into R'_i with entry c'_{ki} . According to, the value of c'_{ki} is smaller than the value of c_{ki} .

$$c'_{ki} = \log_2 (\sqrt{c_{ki}} + 1) \tag{3}$$

Table 2 results from transforming Matrix R_i into R'_i with 18 users and 10 classes by Equation (3). Therefore, Table 1 is transformed into Table 2 through initialization stage.

Table 2. Transformed matrix R'_i

chromo some	Genes in chromosome																	
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}	u_{17}	u_{18}
1	1.45	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0
2	0	0	1.27	0	0	0	0	0	0	0	0	1.27	0	0	1	0	0	1
3	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
10	1	0	1	0	1	1.27	0	1	1	0	0	0	0	0	0	0	1	0

Genetic algorithm offers all objects fitness as a measure to compare function of object with that of other objects or to find out how well an object adjusts itself to virtual environment where Genetic algorithm is applied. Since the calculation of fitness can be based on overall function of object, the most proper fitness calculation function should be applied. This paper calculates fitness to find out classes whose distribution is similar. Therefore, we use Jaccards' method[9]. Equation (4) calculates the similarity of chromosome compared with others. In Equation (4), c'_{ki} indicates the entry in matrix R'_i . c''_{ki} indicates other entries excepting c'_{ki} in matrix R'_i .

$$Fitness(c'_{ki}) = \left(\frac{\sum_{k=1}^n c'_{ki} c''_{ki}}{\sum_{k=1}^n c'^2_{ki} + \sum_{k=1}^n c''^2_{ki} - \sum_{k=1}^n c'_{ki} c''_{ki}} \right) / n \tag{4}$$

In recomposition stage, the value of object function for calculating fitness is recomposed into the other value because the fitness is used as probability. It is called fitness scaling. This paper uses simple selection and Equation (5) for adjusting fitness is used. Equation (5) calculates the ratio of $Fitness(c'_{ki})$ of each class against the total fitness of all chromosomes. According to this, fitness is adjusted and saved in $Fitness\_s(c'_{ki})$.

$$Fitness\_s(c'_{ki}) = \frac{Fitness(c'_{ki})}{\sum_{class=1}^N Fitness(c'_{ik})} \tag{5}$$

Table 3 shows the result of calculating fitness of each chromosome by using Equation (4) and recomposed fitness by using Equation (5).

Table 3. Fitness and recomposed fitness of chromosome

Chromosome	Fitness	Recomposed fitness
1	0.267916	5.61692
2	0.251901	5.281176
3	0.604733	12.67838
4	0.57429	12.04014
5	0.563645	11.81695
6	0.58829	12.33365
7	0.375081	7.863669
8	0.612715	12.84572
9	0.526596	11.04021
10	0.404631	8.483185
Average fitness	0.476	10

In selection stage, chromosome for crossover is selected based on recomposed fitness. In that case, the higher fitness the higher the probability of being selected. Chromosome 3, 4, 5, 6, 8, 9 in Table 3 are highly likely to be selected as parent chromosome. In crossover stage, as for selected document, crossover is conducted according to crossover rate. This paper has used 1-point crossover method. 1-point crossover method is to select one point randomly and to exchange subsequent genes with genes of other classes. P_c is a probability that crossover is conducted on selected class. It is usual that 0.7-0.9 are used as P_c and this paper has used 0.9. In mutation stage, a bit is changed into the other value according to given probability. Mutation rate is probability that genes of chromosome change into another value. Mutation rate frequently uses value ranging from 0.01 to 0.05 and this paper designates 0.01 as mutation rate. Table 4 is chromosome of the second generation as a descendant of the first generation, born after parent chromosome shown in Table 2 went through selection, crossover, and mutation stages.

Table 4. The chromosome and fitness of the second generation

	Gene in chromosome(the second generation)																		Fitness
1	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	0.63157
2	1.27	1.27	1.69	1.58	1.27	1.27	1.45	1	0	1	0	0	0	0	1	1	1.27	1	0.64318
3	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	0.63155
4	1	0	1	0	1.27	0	1	1.45	0	1.45	0	0	0	0	1.45	1.27	1	1.27	0.58190
5	1.27	1.45	1	1.69	1.45	1.45	1.27	1	0	1.45	0	0	0	0	1	1.58	1.45	1	0.62893
6	1.45	0	0	0	1	1	1	1.27	0	0	0	0	0	0	0	0	0	0	0.32592
7	1	1.58	1	1	1	1	1	1.58	0	1	0	0	0	0	1	1	1	1	0.64183
8	1	1	1.27	1	1.27	0	1.27	1	0	1	0	1	1.45	1.45	1.45	1.27	1.27	1.27	0.50842
9	1.27	1.45	1	1.69	1.45	1.45	1.27	1	0	1.45	0	0	0	0	1	1.58	1.45	1	0.62893
10	1	0	1	0	1	1.27	0	1	1	0	0	0	0	0	0	0	1	0	0.35073
	Average fitness=0.553																		

In evaluation stage, whether to continue evolution is decided. If average fitness is larger than or the same as threshold for fitness, evolution is concluded. If it is smaller, evolution repeats from recomposition stage after current chromosome becomes parent. In this paper, we define threshold of fitness to be 1. Since average fitness of chromosome of the second generation calculated in the third column of Table 4 is 0.553 it does not fit in with conclusion condition of Genetic algorithm. Accordingly, evolution to next generation continues. Table 5 demonstrates chromosome from 10 generation to 100 generation with average fitness of each generation.

As in Table. 5, while evolution proceeds average fitness has increased, and in the 100 generation where average fitness becomes 1, evolution has been concluded. User who represents gene not less than 1 in 100 generation is adopted and user that represents gene of 0 is removed. Accordingly, if user that represents gene of 0 at the last generation is removed, a cluster is composed of optimal users. Accordingly, the first cluster is composed of 7 users in Table 5. Excepting users in the first cluster, evolution of population is repeated through initialization, fitness calculation, recomposition, selection, crossover, mutation, and evaluation. Users are grouped into each cluster through these processes. Table 6 shows 18 users grouped into 4 clusters.

Table 5. The chromosome from the first generation to the last generation

Evolution	Chromosome																	Fitness	Average	
10	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	11.6589	0.6573
	1.27	1.27	1.69	1.58	1.27	1.27	1.45	1	0	1	0	0	0	0	1	1	1.27	1	11.2508	
	..																	..		
20	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	11.4441	0.671
	1.27	1.27	1.69	1.58	1.27	1.27	1.45	1	0	1	0	0	0	0	1	1	1.27	1	10.6296	
	..																	..		
30	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	10.9952	0.6877
	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	10.9952	
	..																	..		
40	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	10.9952	0.7332
	1.27	1.27	1.69	1.58	1.27	1.27	1.45	1	0	1	0	0	0	0	1	1	1.27	1	9.97998	
	..																	..		
50	1	0	1	0	1.27	0	1	1.45	0	1.45	0	0	0	0	1.45	1.27	1	1.27	10.9666	0.7632
	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	11.2051	
	..																	..		
60	1	0	1	0	1.27	0	1	1.45	0	1.45	0	0	0	0	1.45	1.27	0	1.27	10.7554	0.8027
	1.27	0	1.58	0	1	1	1	1.27	0	1.27	0	0	0	0	1.27	1.27	1	1.45	11.2535	
	..																	..		
...	...																			
100	1.27	0	1.58	0	0	0	0	1.27	0	1.27	0	0	0	0	1.27	1.27	0	1.45	1	1
	1.27	0	1.58	0	0	0	0	1.27	0	1.27	0	0	0	0	1.27	1.27	0	1.45	1	
	1.27	0	1.58	0	0	0	0	1.27	0	1.27	0	0	0	0	1.27	1.27	0	1.45	1	

Table 6. Clustered users by using Genetic algorithm

Cluster	Users in cluster
1	$u_1, u_3, u_8, u_{10}, u_{15}, u_{16}, u_{18}$
2	u_2, u_4, u_5, u_{17}
3	$u_{11}, u_{12}, u_{13}, u_{14}$
4	u_6, u_7, u_9

4.2 Generating Group Profile

Consider a set of clusters c_i^q with length q. We select feature from grouped users. The feature of grouped users is a set of association words extracted from item. Group profile is a feature vector extracted from grouped users. We define group profile to be as following.

$$C_{i=1}^q = \{C^1, C^2, ..., C^l, ..., C^q\}$$

Profile of *Cluster l* is a feature vector, which is composed of association word. Then *cluster l* is expressed as following. We denote the number of association word in *cluster l* to be r. Each of $\{AW_1, AW_2, ..., AW_r\}$ represents an association word.

$$C^l = \{AW_1^l, AW_2^l, ..., AW_r^l\}$$

Association word is weighted according to the number of features selected from item user rates. Group profile including a weight of association word is as following.

$$C^l = \{k_1^l AW_1^l, k_2^l AW_2^l, ..., k_r^l AW_r^l\}$$

Here, $\{k_1^l, k_2^l, ..., k_r^l\}$ is the weight vector representing a ratio of the number of each association word against the total number of association words in all clusters. Table 7 shows a part of weighted group profile extracted from users in Table 6.

Table 7. Weighted group profile

Cluster	Group profile with weight
1	game&organization&athlete=>match(0.231) sports&participation=>schedule(0.3291)...
2	news&provision&promotion&quick announcemen(0.1938) method&center&manufacturing&utilization(0.8292)...
3	system&business&utilization&method(0.382) world&netizen&person&manipulation&technique(0.281)...
4	contents&site&management&shopping mall(0.2131)

4.3 Discovery of User Preference

New user is finding an item on online. Because new user should be grouped into a cluster as soon as possible, we calculate similarity between new user and group profile and categorize new user into a cluster with the largest similarity. To do this, we compose new user profile. New user profile is composed of features extracted from item. Note that feature is extracted by using Apriori algorithm from item rated by new user, which is interesting to new user if his preference rating for this item is greater than 0.5. In addition, if new user does not rate user preference for an item, feature is extracted from item(document) visited by new user. Therefore, we can solve the first rater problem. In order to improve the accuracy, the weight for an item is appended to user profile. The weight represents the ratio of the number of each association words against all association words selected from interesting items rated by user. We define user profile U_a to be as following. Here, $\{w_1, w_2, \dots, w_t\}$ is the weight vector representing the weight of each association word and t is the number of association words of all kinds in user profile.

$$U_a = \{w_1 \cdot AW_1, w_2 \cdot AW_2, \dots, w_t \cdot AW_t\}$$

In order to group new user to cluster, we use vector similarity. In the field of information retrieval, the similarity between two documents is often measured by treating each document as a vector of word frequency and computing the cosine of the angle formed by the two frequency vectors. We can adopt this formalism to collaborative filtering, where new user profile and group profile take a role of documents, association words take a role of words.

Equation (6) calculates vector similarity between new user profile U_a and each group profile C^i . In Equation (6), the assumption is that the weight of C^i is equal to 0 when association word in group profile is not assigned to user profile and the weight is equal to k_i for the assigned association word.

$$\text{Similarity} (C^i, U_a) = \frac{\sum_{i=1}^t k_i w_i}{\sqrt{\sum_{i=1}^t k_i^2} \sqrt{\sum_{i=1}^t w_i^2}} \quad (6)$$

We assume that there are two web documents, which are interesting to new user, rated or viewed by new user. Table 7 shows features of web documents and class of web document classified by Naïve Bayes classifier.

We construct user profile from features of web document that new user rates. New user profile is shown as follows.

New user profile:

```
{ game&participation&popularity&user&access=>event,
  operation&selection&match&rank&rule=>assessment, television&news=>newspaper
  game&utilization&technology=>development, game&utilization&problem=>rule,
  news&provision=>promotion, announcement&radio=>communication }
```

Table 8. Features and class of web document

Web document	Features	class
1	game&participation&popularity&user&access=>event operation&selection&match&rank&rule=>assessment game&utilization&technology=>development game&utilization&problem=>rule	1
2	news&provision=>promotion announcement&radio=>communication television&news=>newpaper	3

We calculate vector similarity between group profile by each cluster and new user profile using Equation (6). New user is categorized into group whose similarity is the highest. As the result of calculating, new user is categorized into cluster1. Then we recommend web documents in class1 and class3 among web documents rated by users in cluster1 because web documents where new user rates are categorized into class1 and class3. Table 7 shows 7 users in cluster1 and Table 1 shows 15 web documents in class1 and class3 rated by them. We use a frequency of web document in order to gain the weight of user preference for recommending. Each document is sorted by the weight of user preference in order of declining. After all, sorted web documents are recommended for new user.

5 Evaluation

User preference information about web document is obtained from the collaborative filtering database, which consists of 200 customers and 800 different web documents. Users are selected in such a way that they have provided at least 20 ratings among the 800 web documents. 800 Korean Web documents are collected by http down loader from URLs of computer science. Web document has been manually classified into eight classes of computer science. Eight classes is labeled as {game, graphic, news and media, semiconductor, security, Internet, electronic publishing, hardware} and it is expressed as {class1,class2,,,classN} in this paper. Criteria of the classification is statistics of classification that existing information retrieval engines such as AltaVista, Yahoo made on subjects related to computer. Ratings of the first 100 users are used for training while those of the remaining 100 are used for testing.

We use the traditional classification accuracy and the rank scoring metric[5] to evaluate recommendations. The rank scoring metric is an estimate of how likely it is that the user will visit an item on a ranked list. We posit that each successive document in a list is less likely to be viewed by the user with an exponential decay. Each document is sorted by index j in order of declining. Each of $v_{u,j}$, $v_{a,j}$ is a weight of user preference for recommending described in Section 4.3. Equation (7) is the expected utility of a ranked list of documents.

$$R_a = \sum_j \frac{\max(v_{a,j} - d, 0)}{2^{(j-1)/(\alpha-1)}} \tag{7}$$

In Equation (7), d is the neutral vote and α is the viewing halflife. The halflife is the number of items on the list such that there is a 50-50 chance the user will review that document. For these experiments, we use a halflife of 5 documents. The final score reflecting the utilities of all users in the test set is Equation (8).

$$R = 100 \frac{\sum_u R_u}{\sum_u R_u^{\max}} \tag{8}$$

In Equation (8), $R_u^{\max}$ is the maximum possible utility obtained when all documents that user u has accessed appear at the top of the ranked list.

We compare the method of discovery of user preference using Genetic algorithm and Naïve Bayes classifier(G_N_Bayes) with the standard memory based method using Pearson correlation coefficient(P-Corr), model based method using Genetic algorithm(Genetic), and model based method using Naïve Bayes classifier(N_Bayes) varying as the number of users for clustering from 20 to 100.

Fig. 2 shows the accuracy of discovery of user preference. The accuracy of recommendation using G_N_Bayes is 76.6 and is 0.8, 5, 4.2 higher than N_Bayes, Genetic, P-Corr, respectively.

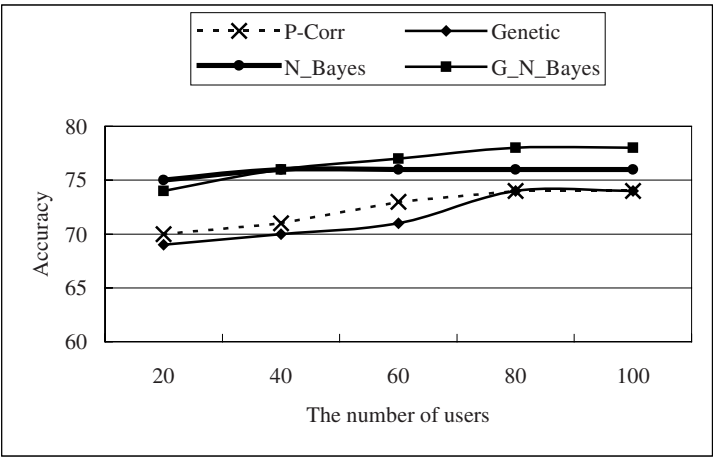


Fig. 2. The accuracy for discovery of user preference

Fig. 3 shows rank score of discovery of user preference. Rank score of recommendation using G_N_Bayes is 61.6 and is 2.2, 6.2, 5 higher than N_Bayes, Genetic, P-Corr, respectively. In addition, Fig. 2 and Fig. 3 show that the larger the number of users, the more performance of G_N_Bayes is generally superior to that of P-Corr and Genetic and N_Bayes.

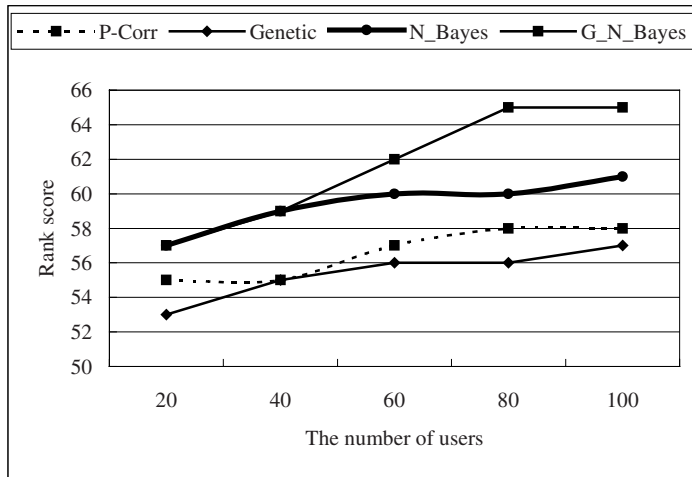


Fig. 3. Rank score for discovery of user preference

6 Conclusions

In this paper, we proposed the method for discovery of user preference using Genetic algorithm and Naïve Bayes algorithm. Genetic algorithm groups user into cluster for collaboration and Naïve Bayes algorithm categorizes web documents for content based recommendation. Hybrid collaborative recommendation and content based recommendation outperforms traditional collaborative recommendation and the recommendation using Genetic algorithm and the recommendation using Naïve Bayes algorithm.

In the future, we would like to verify this method through more users and combine proposed method with memory based method.

References

1. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.
2. R. Agrawal and T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," In Proceedings of the 1993 ACM SIGMOD Conference, Washington DC, USA, 1993.
3. C. Basu and H. Hirsh and W. W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation," In proceedings of the Fifteenth National Conference on Artificial Intelligence, pp. 714-720, Madison, WI, 1998.
4. D. Billsu and M. J. Pazzani, "Learning collaborative information filters," In proceedings of the Fifteenth International Conference on Machine Learning, 1998.

5. John. S. Breese and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998.
6. Y. H. Chien and E. I. George, "A Bayesian Model for Collaborative Filtering," In Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, San Francisco, 1999.
7. M. O. Connor and J. Herlocker, "Clustering Items for Collaborative Filtering," Proceedings of the ACM SIGIR Workshop on Recommender Systems, Berkeley, CA, 1999.
8. J. Delgado and N. Ishii, "Formal Models for Learning of User Preferences, a Preliminary Report," In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden, July, 1999.
9. M. Gondon, "Probabilistic and Genetic algorithms for document retrieval," Communication of the ACM, 31, pp. 1208-1218, 1988.
10. G. Karypis, "Evaluation of Item-Based Top-N Recommendation Algorithms," Technical Report CS-TR-00-46, Computer Science Dept., University of Minnesota, 2000.
11. S. J. Ko and J. H. Lee, "Optimized Association Word Knowledge Base Construction using Apriori-Genetic Algorithm," Proceedings of the Third International Symposium on Adaptive Systems (ISAS2001), pp. 29-36, 2001.
12. S. J. Ko and J. H. Lee, "Feature Selection using Association Word Mining for Classification," In Proceedings of the Conference on DEXA2001, LNCS2113, pp. 211-220, 2001.
13. D. D. Lewis, *Representation and Learning in Information Retrieval*, PhD thesis (Technical Report pp. 91-93, Computer Science Dept., Univ. of Massachusetts at Amherst, 1992.
14. T. Michael, *Maching Learning*, McGraw-Hill, pp. 154-200, 1997.
15. D. Mladenic, "Feature subset selection in text-learning," Proceedings of the 10th European Conference on Machine Learning, pp. 95-100, 1998.
16. M. Pazzani, D. Billsus, *Learning and Revising User Profiles: The Identification of Interesting Web Sites*, Machine Learning 27, Kluwer Academic Publishers, pp. 313-331, 1997.
17. D. M. Pennock and E. Horvitz, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory-and Model-Based Approach," Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000.
18. R. Raymond and J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio, TX, pp. 195-204, June, 2000.
19. V. Rijsbergen and C. Joost, *Information Retrieval*, Butterworths, London-second edition, 1979.
20. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.
21. B. M. Sarwar, J. A. Konstan, Al Borchers, J. Herlocker, B. Miller, and J. Riedl, "Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System," Proceedings of the 1998 Conference on Computer Supported Cooperative Work. Nov. 1998.
22. L. H. Ungar and D. P. Foster, "Clustering Methods for Collaborative Filtering," In Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence, 1998.
23. Introduction to Rainbow URL: <http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naïve-bayes.html>.

Using the Quantum Data Model to Develop Shareable Definitions

Harry Ellis

British Army

Army Data Services, Faraday Building, Blandford Camp

BLANDFORD FORUM, Dorset, DT11 8RE,

UK

harryellis@compuserve.com

Abstract. This paper contains an outline of a new form of data model with particular focus on the way real-world objects are classified. Concepts such as categorization schemes and designation schemes that are widely used in the real world are made available for conceptual modeling in a new language called CBML. Case material is used to show how this new language can help to reconcile different perceptions of information. In particular it is shown how this can lead to an enterprise model with formal links to those of specific business areas and information systems.

1 Introduction

This paper introduces a new form of conceptual modeling that has emerged from practical experience within the British Army. This development is a result of extensive research by the author and colleagues within Army Data Services (ADS). This research was undertaken while striving to create a large conceptual schema with the clarity and rigor needed to support the sharing of information across this large and complex enterprise.

1.1 Technical Development Driven by Necessity

Data sharing is an imperative for the British Army. It has a massive legacy of disparate systems, information needs that are changing all the time and a goal of “fully interoperable information systems by 2010”. It was found that some of the difficulties faced were due to deficiencies in the languages available for conceptual modeling. For this reason ADS decided that it had to take a fresh and critical look at the concepts, methods and tools that are currently available for conceptual modeling.

1.2 What Is the Quantum Data Model?

The *quantum data model* is a set of concepts for use in defining the structure and meaning of information. It has been devised by ADS as a necessary step towards the Army’s interoperability goal. It differs from the currently dominant *relational model* by a shift of focus away from tables that represent types of real-world object and toward the various types of information that are used for their description.

This new set of concepts can be seen as a synthesis of ER and the data aspects of UML together with some key innovations. The roots of this radical development lie in difficulties arising from the limited forms of generalization that underpin all the established languages for conceptual modeling. It was found that these difficulties could only be overcome with the addition of certain new concepts. The name *quantum data model* was chosen because these new concepts describe an enterprise in terms of "the superposition of all states of various types of information about real-world objects of various types". This echoes the general idea of "the quantum description of a system".

1.3. What Is New about It?

All widely used languages for conceptual modeling of information are based on the notion that each real-world object can be represented as an instance of a single entity class. It is further assumed that this entity class should determine the nature of the attributes and relationships by which real-world objects of that type are described. In ADS we found that real-world objects are not that simple. The currently dominant approach to conceptual modeling leads to attributes and relationships that are tainted with thinking that is more in tune with data design than with the way in which information is perceived in the real world.

The *quantum data model* contains three important technical ideas that allow the most complex information to be defined in a consistent way by simple unambiguous statements. We found that dialogue with end users works best when focused on discrete elements of information that are self-contained units of change. Some of these are simple and could be represented by an attribute or foreign key. Others are quite complex and would need a whole table if normalization rules were strictly applied. We found that real-world objects can be categorized in many different ways and that in many cases this categorization may vary over time.

We also found that the many entity classes involved in a substantial enterprise need to be grouped into sets within which certain rules apply. Some of these entity class sets serve as *designation schemes* used to classify real-world objects when they first become of interest. Other entity sets known as *categorization schemes* provide a rational framework for subsequent further categorization.

1.4 Why Was This Development Necessary?

The development of innovative information technology is not a core function for either the British Army or its data services organization. For this reason every endeavor was made to meet the Army's need for an enterprise level conceptual schema with well established modeling techniques. The prime candidates for this task were ER (Barker) and UML but both of these proved unsuitable for a conceptual schema that must be able to grow and adapt with changing business needs. It was only when it became apparent that these languages were causing more problems than they solved that we were forced into the fundamental thinking that led to the ideas of the *quantum data model*.

Having developed this new body of concepts it was necessary for them to be formalized into a language with support tools and training. The name "Corporate Business Modelling Language" (CBML) is derived from its *corporate* (enterprise)

viewpoint and focus on *business* (real-world) objects. For the remainder of this paper, discussion of the general principles of the quantum data model is illustrated through the specific terminology and graphical notation of CBML. Full definition of this language can be obtained from the website: www.CBML.co.uk [1].

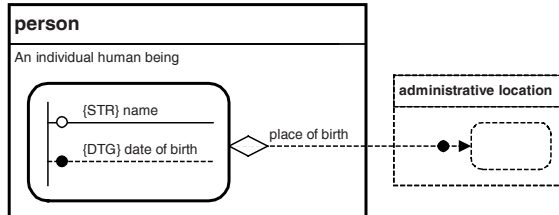
2 Key Features of CBML

The key difference between CBML and the established languages for conceptual modeling is a shift of focus from entity classes on to individual *characteristics*. These characteristics are elementary units of information that have meaning on their own but cannot be split in any way without destroying that meaning.

The effect of this shift of focus is profound. It enables us to say categorically that every entity class represents a type of real-world object and never represents a type of information. With this new focus it becomes possible for the modeler to formally distinguish types of information from the types of real-world object that the information describes, i.e. to distinguish between the content and the subject.

2.1 Characteristics

Information is represented in CBML by *characteristics* while subjects are represented by entity classes. The function of the conceptual schema is now easy to explain. It says which types of content (characteristics) are used to describe which types of real-world object (entity classes).



Each instance of **person** is an individual human being described by:
 "name" (mandatory, variable, character string)
 "date of birth" (optional, fixed, date-time)
 "place of birth" (optional, fixed, ref to single instance of administrative location)

Fig. 1. Example of CBML graphical notation

This first glimpse of CBML in Figure 1. illustrates the basic style of its graphical notation. It shows that the entity classes can be modeled individually and that reference can be made to another entity class without knowing anything about it beyond its name. The most striking visual feature of CBML is its use of a double box. The outer box (deliberately reminiscent of a UML class) represents the entity class as such, e.g. the concept of "person". The inner box (deliberately reminiscent of an ER (Barker) entity soft box) represents the instances, e.g. unspecified individual persons.

The reason for drawing this distinction is to aid verbalization and diagram readability. The model expresses rules. Some of these rules are expressed as a

definition of the class as such. A good example of this is the generalization concept in UML. Other types of rule are associated with the class but are expressed as applicable not to the class as such but to its members (instances). A good example of this is the verbalization of relationships in ER (Barker). The double box notation enables a CBML modeler to attach class definitions to the outer 'class' box while rules that apply to its members are attached to or placed inside the inner box. We have found that this makes it easier for user representatives to read the models and appreciate their full implications. In visual terms this gives the best of both worlds - an ER model embedded inside a UML class diagram.

2.2 History, Projection, and Competing Versions

One of the key problems faced in our modeling was to decide which characteristics needed history, projection and/or allowance for competing versions. It was found impossible to say for any characteristic that these would never be required. This threatened to make the model both complex and subjective. To avoid this we decided that the model should incorporate an assumption that the implementation would provide for history, projection and competing versions for all characteristics. This underlying provision could then be invoked as operational circumstances required. This is a big assumption but is arguably sound since the usual presumption of 'latest value only' is just a limiting case and would be covered anyway.

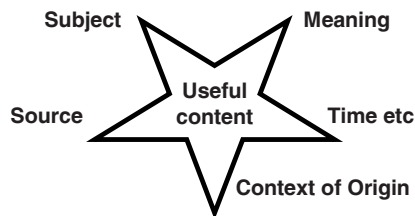


Fig. 2. Information Elements

Figure 2 illustrates the core idea of quantization. It is this principle that has enabled ADS to provide an implementation vehicle that deals with history, projection and competing versions in a comprehensive and thus economical way. Each element of information is stored such that its *useful content* is forever associated with a control wrapper that serves as a sort of passport that enables it to reside, move and be copied anywhere in cyberspace. This 'passport' contains five vital pieces of control information.

Subject. This identifies the real-world object that is described. It does this by citing the globally unique identifier of the real-world object concerned.

Meaning. This identifies the relevant characteristic as defined in the conceptual schema.

Source. This is a pointer that identifies the person, organization or information process that created the useful content.

Context of Origin. This identifies the operational situation, activity, geographic location etc to which the useful content is relevant.

Time Etc.. This refers to a set of qualifiers that, together with a globally unique identifier, serve to distinguish different versions from the same source with the same context of origin such as a locus of positions for a moving vehicle.

2.3 Complex Characteristics

It was said earlier that all characteristics for any given real-world object must each have their own independent life history comprising a set of *information elements*. This means that the ‘useful content’ must have a structure that is appropriate for the characteristic concerned. Some characteristics like ‘name’ are easy. The ‘useful content’ is a single character string. It is not always that simple. If the self-contained unit of information represented by a characteristic is more complex than any of the standard data types of ‘string’, ‘number’ etc. then an appropriate abstract data type must be created.

2.4 Categorisation

The new type of *characteristic* known as *categorization* is needed to show clearly where information about a real-world object implies additional typing with associated inheritance of *characteristics*. This type of characteristic is defined as a reference to a set of *abstract entity classes* representing a choice of categories. *Characteristics* defined for these abstract entity classes are inherited through the *categorization* but only while it is in force.

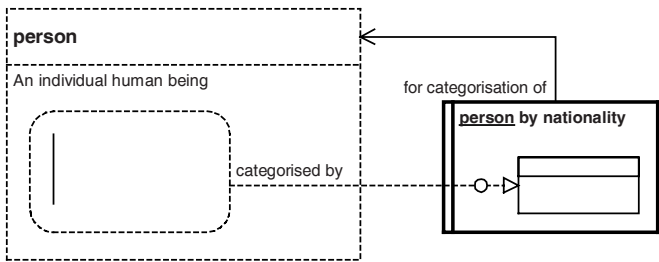
This concept of categorization is of great importance. What we are talking about here is usually modeled as either an attribute from an enumerated domain or a reference to an entity class that represents a set of type definitions. We have found that both of these concepts are deeply flawed when used in a conceptual schema. It is recognized that this is a non-trivial issue. Work is in hand to address the issue of generalization in the conceptual schema in a full research paper. The point being made here is that the concept of categorization is well understood and widely applied in the real world and for this reason must be recognized formally within any language used for the expression of a conceptual schema. By ‘well understood’ is meant that people in general are inclined to presume that if “A is a B” then everything known about B must apply to A. This much is intuitive and may be assumed by user representatives wherever any detail in a conceptual schema can be read or interpreted as a statement in the form “A is a B”.

2.5 Entity Class Sets

In languages currently used for conceptual modeling the only way in which entity classes can be grouped is as sub-types of a more generic entity class. The problem with this is that across an enterprise there may be many such sub-types that have little to do with each other. Suppose we have four sub-types of ‘vehicle’: ‘land vehicle’, ‘sea vehicle’, ‘military vehicle’ and ‘civilian vehicle’. In the real world these are recognized as falling into two groups: ‘vehicle by mobility’ and ‘vehicle by military/civilian status’. To classify an instance of vehicle it is necessary to pick one

type from each group. Groups like this are represented in CBML by sets of entity classes known as *designation schemes* or *categorization schemes* according to the purpose for which they are defined.

The following diagram shows how a *categorization scheme* “person by nationality” would be expressed in CBML. Note that two relationships are involved here, each being a distinct form of generalization. First there is the statement that the *categorization scheme* “person by nationality” is *for categorisation of* persons. This says formally that members of this scheme such as ‘British person’, ‘Japanese person’ etc. are all types of person and must not be used to categorize any real-world object unless it is already known to be a person of sorts.



"**person by nationality** is a categorization scheme whose members may be cited as categorisation of instances of **person**"

"Each instance of **person** may be categorised by reference to one variable member of the categorization scheme **person by nationality**"

Fig. 3. Categorization Scheme

The second statement represented by the broken line labeled ‘categorized by’ is a rule for generalization of instances of person. In the case illustrated here the rule says that categorization of persons by nationality is optional and variable but allows only one category at a time. This would exclude dual nationality and thus represents a rather narrow view. In CBML it would be easy to alter this to indicate the possibility of multiple nationality by using a double-headed arrow.

2.6 Designation Scheme

This example shows a designation scheme called “military person by Army designation” whose members are concrete entity classes. Such schemes are found wherever a business area has devised its own set of type designations each of which incorporates a distinctive name and a particular combination of further categorizations. In the case illustrated, the Army is shown as having its own set of type designations for military person where each represents a specific combination of service and rank e.g. ‘army major’ and ‘navy captain’. When a real-world object is created as an instance of the type designation ‘army major’ it is immediately and permanently defined as inheriting applicability of the characteristics defined for the four entity classes: ‘person’, ‘military person’, ‘army person’ and ‘major person’.

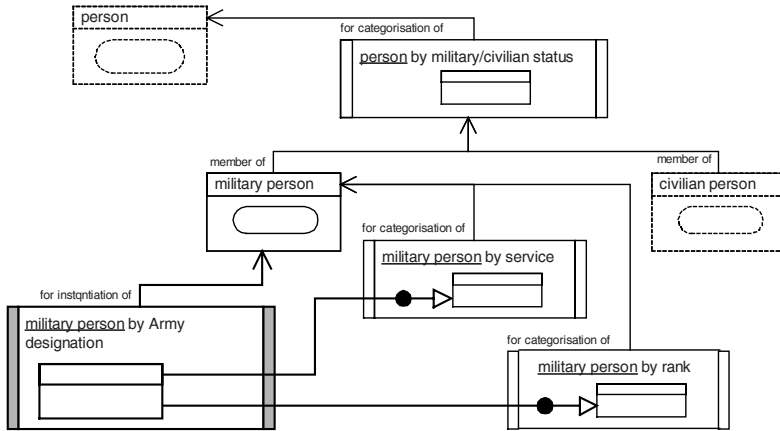


Fig. 4. Designation Scheme

3 Comparison with Other Languages

3.1 Why Not Use ER?

ER modeling is designed to identify “the things of importance in an organisation, the properties of those things and how they are related to one another” [2]. It has simplistic inheritance structures and it relies on the modeler to choose between alternative modeling constructs. The only reliable definition of the entity concept is as something that can usefully be represented by a data table. For this reason the language is suited more to ‘logical’ than ‘conceptual’ modeling. The simplicity of the language makes it easy to use but very difficult to use well. The creation of entity models that can be read accurately by persons other than their creators demands a combination of high skill and inventiveness with rigorous control of style. This combination is largely unobtainable in the number of modelers needed by a large enterprise.

3.2 Why Not Use UML ?

“UML is a graphical language for visualizing, specifying, constructing and documenting the artifacts of a software intensive system” [3]. This categorical statement from the originators of UML provides a fair indication of the unsuitability of UML as a language for the construction of a conceptual schema. One of its great strengths is the inheritance structure but this is more rigid than those used in the real world. Another strength lies in the process methods but these are of little, if any, relevance to the aims of a conceptual schema. The problem here is that the real world of an enterprise does not behave like a single rule-based machine. It is more like a federation of active resources intent on pursuing individual goals through collaboration. This cannot be described in UML. At a more superficial level it is

found that both verbal and graphic representation of associations is poor. There is also the problem that data hiding which is great as an implementation device is inappropriate for the definition of information that needs to be shared.

3.3 Why Not Use ORM ?

ORM “is a semantic modeling approach that views the world simply in terms of objects playing roles (parts in relationships)” [4]. ORM is aimed directly at the conceptual schema and its verbalization is a strong feature. The graphics are rather cumbersome and play a relatively minor role in the endorsement by user representatives. The main problem is that the verbalized facts can lead both modeler and user to read far more into a model than is formally defined. This derives from the great reliance on its similarity with natural language and wide choice of expression. It also suffers from the same limited form of generalization as UML.

3.4 Why Is CBML Better ?

The above criticisms of ER, UML and ORM cannot be justified here for want of space. The aim here is not to condemn the use of these languages for conceptual modeling but to introduce CBML and explain how and why it is different. The main strength of CBML derives from its being designed to define “types of real world object and the types of information by which they are described”. It is for this reason that it has formal ways of expressing multi-way sub-typing and transient inheritance as used in the real world. The clear distinction between type and instance means that the need for creative modeling can be eliminated. This means that the modeler can concentrate on the analysis of information needs and their expression in a standard way that all-comers can read with the same understanding. Another major strength is the ability to define an entity class to include references to other entity classes whose own characteristics are as yet undefined. This means that models can grow and adapt without rework.

4 Data Standardisation in a Large Enterprise

The main driver for the development of CBML was interoperability of information systems. This requires standardization at best and reliable two-way mapping at worst. It is hoped with some justification that CBML will facilitate the creation of enterprise definitions that are truly independent of implementation choices. These can then be implemented in various ways in the certain knowledge that the information represented is meant to be the same and is properly defined in the conceptual schema.

To achieve this the language must be supported by a method that can cope with a pattern of application development in which progress is made at different rates at different times in different parts of the enterprise. This means that legacy data is always important. The method developed around CBML is based on an awareness that a conceptual model must satisfy two conflicting aims. To be properly endorsed it must relate directly to user perceptions while for implementation it needs technical precision. In CBML this gulf is bridged by formal recognition of five distinct levels of model. These levels provide a disciplined framework within which different perceptions of information can be reconciled. The five levels of information model are set out below.

4.1 Legacy Models

Legacy models are at their most important in the early stages of establishing a conceptual schema for the enterprise. There are two distinct levels as follows.

Actual Legacy Models (Level A). Models at this level are a faithful representation of actual data as it exists. The relational data model is a form commonly found at this level.

Baseline legacy Models (Level B). Each of these is an adaptation of one *actual legacy model* in which each possible interpretation of each type of information has been given its own distinct representation and name.

4.2 CBML Models

CBML models are genuine conceptual models that may differ in purpose, scope and quality as follows.

Candidate CBML Models (Level C). These are models defined in CBML that express the information needs of a specific business area in terms that reflect current user perceptions.

Developed CBML Models (Level D). Each of these is a developed version of a specific *candidate CBML model* that has been refined in accordance with CBML best modeling practice but without consideration of any needs that lie outside the scope of the business area concerned.

Enterprise CBML Models (Level E). each of these covers the information needs that are common across a complete enterprise. As such it is a synthesis of all relevant *developed CBML models*.

Comparison of Levels. All CBML models are required to conform fully to the definitions and graphical notation. However, for *candidate CBML* models the overriding consideration is the ease with which the people who use the information can understand, review and endorse the model. It follows that the definitions of *characteristics* should be simple and direct even though this may fall short of best modeling practice in some respects. It is in the progress to level D that the precision needed for implementation is applied. This is a prerequisite for any attempt to reconcile the conflicting needs of different business areas in order to create agreed definitions for inclusion at level E.

5 Case Study

The following illustrations are taken from actual modeling experience on an aspect of army information which is of concern to two distinct business areas. One of these is

the central personnel function. The other area covers the individual fighting and support units where the immediate information needs of these units are heavily influenced by a major legacy system known as UNICOM. The aim of this case study is to show how CBML helps with the reconciliation of different perceptions of common information.

5.1 Development of Conceptual Schema for a Business Area

The model shown in Figure 5 shows the way in which the concepts of “person” and “military person” were initially perceived by the business area for personnel. This is a *candidate model* (level C).

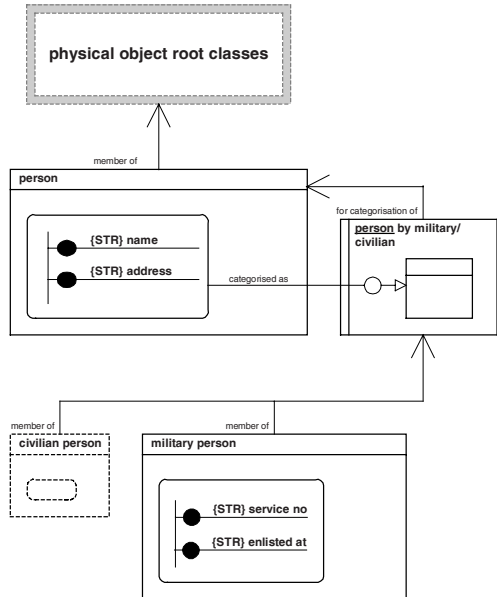


Fig. 5. Candidate Model – Personnel Area

This model says that each person must have fixed string characteristics for ‘name’ and ‘address’ and a variable categorization as either military or civilian. Variability is indicated by the blobs which are solid for ‘fixed’ and hollow for ‘variable. The single hollow arrowhead means ‘categorization by reference to only one member category at a time.

The concept of ‘military person’ is shown as an abstract entity class and member of the categorization scheme “person by military/civilian”. This states formally that all instances of ‘military person’ inherit, from the entity class ‘person’, applicability of the characteristics ‘name’ and ‘address’. In addition to this inheritance there are two further string characteristics called ‘service number’ and ‘enlisted at’.

In the developed version shown in Figure 6 the representation of the characteristic ‘enlisted at’ has been changed from a string to a referential characteristic that is mandatory, fixed and refers to just one instance of a new entity class called ‘recruiting

office'. This new entity class is shown with broken outline to indicate that it is not fully defined on this diagram and may not yet be defined anywhere.

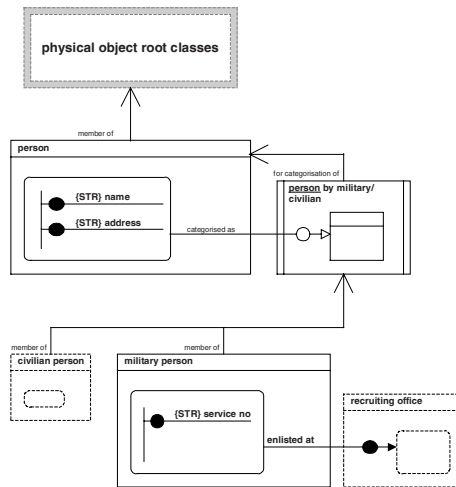


Fig. 6. Developed Model – Personnel Area

5.2 Development of Conceptual Schema for a Legacy System

Figure 7 shows a candidate model for part of the UNICOM system. This shows three entity classes each based on a table within the existing database.

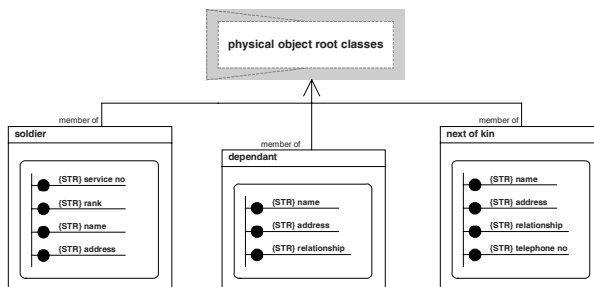


Fig. 7. Candidate Model – UNICOM System

It is immediately evident from this diagram that this is somewhat naïve for a conceptual schema since all of the characteristics are mandatory, fixed character strings. The improved version shown in figure 8 reflects a recognition that the concepts of 'soldier', 'dependent' and 'next of kin' are all roles of 'person'.

Further development based on application of rules for good modelling practice has the characteristic 'rank' changed into a mandatory but variable categorization

reference to any one member of a new *categorization scheme* called ‘army person by rank’. This is illustrated in figure 8.

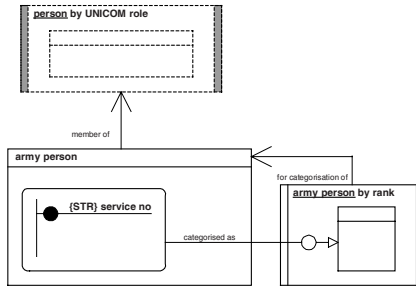


Fig. 8. Developed Model – UNICOM System

5.3 Enterprise Model

When the developed models of the Personnel Area and the UNICOM system are compared it is clear that the more generic concepts of person with name, address and categorization by military/civilian are shared by both. Figure 9 shows how this is reflected in the enterprise model

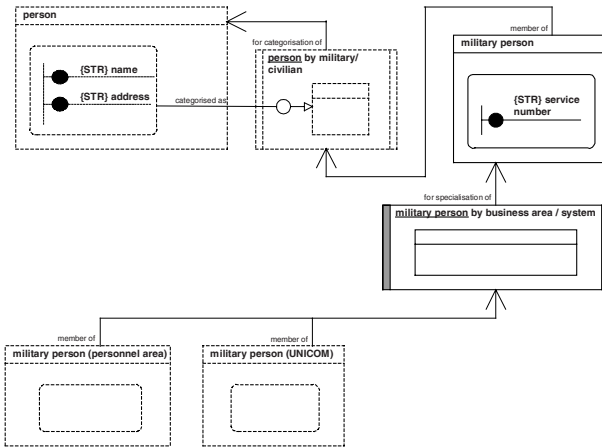


Fig. 9. Enterprise Model

A *designation scheme* is introduced with the name “military person by business area / system”. This has member entity classes called ‘military person (personnel area)’ and ‘military person (UNICOM)’ that are fully defined only within the developed models for the respective business area / system. We now have a situation in which there are specialized versions of military person that inherit from the enterprise model the common concept of military person with the characteristic ‘service number’ and further inheritance of the concept of person with its characteristics ‘name’ and ‘address’.

6 Conclusion

This brief introduction to CBML has attempted to explain why one major enterprise has seen fit to question and improve on the established languages for conceptual modeling. The new concepts described here have emerged from a determination to breathe life into the idea of a conceptual schema in the context of a large and complex enterprise with a growing need for the sharing of information.

While the thinking behind this development owes much to the pressures faced by the British Army the impact of global communications is applying similar pressures in other fields. It is hoped that the wider research community will consider these ideas to evaluate their technical merit and potential for wider application.

References

1. Army Data Services: Corporate Business Modelling Language (CBML) Baseline (Version 2.0), Available from <http://www.cbml.co.uk> published by Army Data Services, Faraday Building, Blandford Camp, Blandford Forum, Dorset, DT11 8RE, UK (2001).
2. Barker, R.: Case*Method Entity Relationship Modeling, Addison-Wesley Publishing Company (1990).
3. Rumbaugh, J., Jacobson, I. and Booch, G.: The Unified Modeling Language Reference Manual, Addison-Wesley (1999).
4. Halpin, T.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design, Morgan Kaufmann Publishers (2001).

Author Index

- Arai, Eiji 57, 141
Arisawa, Hiroshi 105, 115
Atzeni, Paolo 253
- Barker, Ken 278
Berkers, Frank 91
Bernauer, Martin 363
Bovenkamp, Martin Van De 67
Bussler, Christoph 377
- Cáceres, P. 227
Cappi, Juan 457
Casteleyn, Sven 198
Cavero, J.M. 227
Chang, Jae-Woo 320
Chikano, Machiko 57
Crescenzi, Valter 264
- Dobbie, Gillian 292
- Eijnatten, Frans M. van 1, 67
Ellis, Harry 485
Embley, David W. 212
Ermolayev, Vadim 390
- Fortier, Andrés 457
Frank, Ulrich 444
Fröhlich, Peter 166
- Goossenaerts, Jan 91, 127
Grote, Gudela 67
- Hammer, Dieter 91
Hatano, Kenji 334
Hiraide, Yasuhiko 57
Hu, Zaijun 166
- Imai, Sayaka 105
Itoh, Kiyoshi 30
- Janowski, Tomasz 432
Jayaweera, Prasad 417
Johannesson, Paul 417
Jongkind, Ruben 67
- Kamimura, Manabu 30
Kappel, Gerti 183
Karlalalem, Kamalakkar 348
Kato, Naoyuki 30
Kawabata, Ryo 30
Keberle, Natalya 390
Khatri, Vijay 152
Kinutani, Hiroko 334
Ko, SuJeong 471
Krishnan, Rupa 348
Kumagai, Satoshi 30
Kurosu, Noriaki 44
- Lausen, Georg 306
Lawrence, Ramon 278
Lechner, Stephan 239
Lee, JungHyun 471
Lee, Mong Li 292
Lee, Sin Yeung 292
Lehtelä, Jouni 67
Leskinen, Timo 67
Liddle, Stephen W. 212
Ling, Tok Wang 292
Little, Scott 67
- Marcos, E. 227
Mecca, Giansalvatore 264
Merialdo, Paolo 253, 264
Munaga, Lalitha 348
- Noro, Kageyu 81
- Ojo, Adegboyega 432
- Pelletier, Christine 127
Preuner, Günter 239
Pröll, B. 183
- Ram, Sudha 152
Retschitzegger, Werner 183
Rhijn, Gu Van 67
Rossi, Gustavo 457
- Sato, Takako 115
Schoelzke, Manfred 166
Schrefl, Michael 239, 363
Schwabe, Daniel 457
Schwinger, Wieland 183

Shirase, Keiichi 141
 Sindoni, Giuseppe 253

Tanaka, Ryohei 81
 Taveter, Kuldar 403
 Tolok, Vyachyslav 390
 Tomii, Takashi 105, 115
 Tomita, Yoshitaka 57
 Troyer, Olga De 198
 Tsumaya, Akira 141

Uemura, Shunsuke 334

Vela, B. 227
 Vink, Peter 1, 67

Wäfler, Toni 67
 Wagner, Gerd 403
 Wakamatsu, Hidefumi 141
 Wei, Fang 306
 Wohed, Petia 417
 Wortmann, Hans 91
 Wu, Xiaoying 292

Yamada, Shunji 10, 44
 Yau, Sai Ho 212
 Yoshikawa, Masatoshi 334

Zeng, Daniel Dajun 152
 Zhang, Limin 152