Yannis Manolopoulos
Joaquim Filipe
Panos Constantopoulos
José Cordeiro (Eds.)

Enterprise Information Systems

LNBIP 3

8th International Conference, ICEIS 2006
Paphos, Cyprus, May 2006
Revised Selected Papers

Springer

Lecture Notes
in Business Information Processing 3

Yannis Manolopoulos Joaquim Filipe
Panos Constantopoulos José Cordeiro (Eds.)

Enterprise Information Systems

8th International Conference, ICEIS 2006
Paphos, Cyprus, May 23-27, 2006
Revised Selected Papers

Springer

Volume Editors

Yannis Manolopoulos
Aristotle University
54124 Thessaloniki, Greece
E-mail: manolopo@csd.auth.gr

Joaquim Filipe
Polytechnic Institute of Setúbal / INSTICC
2910-761, Setúbal, Portugal
E-mail: j.filipe@est.ips.pt

Panos Constantopoulos
Athens University of Economics and Business
Athens, Greece
E-mail: panos@ics.forth.gr

José Cordeiro
Polytechnic Institute of Setúbal / INSTICC
2910-761 Setúbal, Portugal
E-mail: jcordeiro@est.ips.pt

Library of Congress Control Number: 2007943480

ACM Computing Classification (1998): J.1, H.4, D.2, I.2.1

ISSN 1865-1348
ISBN-10 3-540-77580-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-77580-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12211241 06/3180 5 4 3 2 1 0

Preface

This book contains the best papers of the 8th International Conference on Enterprise Information Systems (ICEIS 2006), held in the city of Paphos (Cyprus), organized by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) in collaboration with the University of Cyprus, Aristotle University of Thessaloniki and Athens University of Economics and Business.

ICEIS has become a major point of contact between research scientists, engineers and practitioners in the area of business applications of information systems. This year, five simultaneous tracks were held, covering different aspects related to enterprise computing, including: "Databases and Information Systems Integration," "Artificial Intelligence and Decision Support Systems," "Information Systems Analysis and Specification," "Software Agents and Internet Computing" and "Human–Computer Interaction." All tracks focus on real world applications and highlight the benefits of Information Systems and Technology for industry and services, thus making a bridge between Academia and Enterprise.

Following the success of 2005, ICEIS 2006 received 404 paper submissions from more than 40 countries spanning all continents. In all, 63 papers were published and presented as full papers, i.e., completed work (8 pages in proceedings / 30–min oral presentations), 102 papers reflecting work-in-progress or position papers were accepted for short presentation and another 75 for poster presentation.

These numbers, leading to a "full-paper" acceptance ratio below 16%, and a total acceptance ratio including posters below 60%, show the intention of preserving a high-quality forum for the next editions of this conference. Additionally, as usual in the ICEIS conference series, a number of invited talks, including keynote lectures, case studies and technical tutorials were also held. These special sessions, presented by internationally recognized specialists in different areas, definitely contributed to increasing the overall quality of the conference and to providing a deeper understanding of the enterprise information systems field.

The program for this conference required the dedicated effort of many people. Firstly, we must thank the authors, whose research and development efforts are recorded here. Secondly, we thank the members of the Program Committee and the additional reviewers for their diligence and expert reviewing. Thirdly, we thank the invited speakers for their invaluable contribution and for taking the time to synthesize and prepare their talks. Fourthly, we thank the workshop Chairs whose collaboration with ICEIS was much appreciated. Finally, special thanks to all the members of the Local Organizing Committee, especially Panicos Masouras, whose collaboration was fundamental for the success of this conference.

October 2007

Yannis Manolopoulos
Joaquim Filipe
Panos Constantopoulos
José Cordeiro

Conference Committees

Conference Co-chairs

Yannis Manolopoulos, Aristotle University of Thessaloniki, Greece
Joaquim Filipe, Polytechnic Institute of Setúbal / INSTICC, Portugal

Program Co-chairs

Panos Constantopoulos, Athens University of Economics and Business, Greece
José Cordeiro, Polytechnic Institute of Setúbal / INSTICC, Portugal

Organizing Committee

Paulo Brito, INSTICC, Portugal
Helder Coelhas, INSTICC, Portugal
Bruno Encarnação, INSTICC, Portugal
Vitor Pedrosa, INSTICC, Portugal
Mónica Saramago, INSTICC, Portugal

Senior Program Committee

Luís Amaral, Portugal
Peter Bøgh Andersen, Denmark
Ricardo Baeza-Yates, Chile
Senén Barro, Spain
Jean Bézivin, France
Enrique Bonsón, Spain
João Alvaro Carvalho, Portugal
Albert Cheng, USA
Bernard Coulette, France
Jan Dietz, The Netherlands
Frank Dignum, The Netherlands
Schahram Dustdar, Austria
António Figueiredo, Portugal
Ulrich Frank, Germany
Göran Goldkuhl, Sweden
Nuno Guimarães, Portugal
Jatinder Gupta, USA
Jean-Paul Haton, France
Dimitris Karagiannis, Austria
Maurizio Lenzerini, Italy
Michel Leonard, Switzerland
Kecheng Liu, UK
Pericles Loucopoulos, UK
Paul Luker, UK
Kalle Lyytinen, USA
Yannis Manolopoulos, Greece
José Legatheaux Martins, Portugal
Masao Johannes Matsumoto, Japan
Luís Moniz Pereira, Portugal
Alain Pirotte, Belgium
James Odell, USA
George Papadopoulos, Cyprus
Marcin Paprzycki, USA
Klaus Pohl, Germany
Matthias Rauterberg, The Netherlands
Colette Rolland, France
Narcyz Roztocki, USA
Abdel-Badeeh Salem, Egypt
Bernadette Sharp, UK

Timothy K. Shih, Taiwan
Alexander Smirnov, Russia
Ronald Stamper, UK
Miguel Toro, Spain
Antonio Vallecillo, Spain
Michalis Vazirgiannis, Greece
François Vernadat, Luxembourg
Ioannis Vlahavas, Greece
Frank Wang, UK
Merrill Warkentin, USA
Hans Weigand, The Netherlands
Yoneo Yano, Japan

Program Committee

Patrick Albers, France
Salah Al-Sharhan, Kuwait
Andreas Andreou, Cyprus
Pedro Antunes, Portugal
Juan Carlos Augusto, UK
Bart Baesens, UK
Cecilia Baranauskas, Brazil
Steve Barker, UK
Balbir Barn, UK
Nick Bassiliades, Greece
Remi Bastide, France
Nadia Bellalem, France
Hatem Ben Sta, Tunisia
Peter Bertok, Australia
Robert Biddle, Canada
Oliver Bittel, Germany
Luis Borges Gouveia, Portugal
Djamel Bouchaffra, USA
Danielle Boulanger, France
Jean-Louis Boulanger, France
José Ângelo Braga de Vasconcelos, Portugal
Miguel Calejo, Portugal
Coral Calero, Spain
Luis M. Camarinha-Matos, Portugal
Jorge Cardoso, Portugal
Fernando Carvalho, Brazil
Jose Jesus Castro-Schez, Spain
Luca Cernuzzi, Paraguay
Maria Filomena Cerqueira de Castro Lopes, Portugal
Laurent Chapelier, France
William Cheng-Chung Chu, Taiwan
Chrisment Claude, France
Francesco Colace, Italy
Cesar Collazos, Colombia
Antonio Corral, Spain
Julie Cowie, UK
Sharon Cox, UK
Mohamed Dahchour, Morocco
Sergio de Cesare, UK
Andrea De Lucia, Italy
Nuno de Magalhães Ribeiro, Portugal
José Javier Dolado, Spain
Alan Eardley, UK
David Emery, UK
Jean-Max Estay, France
João Faria, Portugal
Eduardo Fernández-Medina, Spain
Edilson Ferneda, Brazil
Paulo Ferreira, Portugal
Filomena Ferrucci, Italy
Andre Flory, France
Donal Flynn, UK
Ana Fred, Portugal
Lixin Fu, USA
Juan Garbajosa, Spain
Marcela Genero, Spain
Joseph Giampapa, USA
Raúl Giráldez, Spain
Pascual González, Spain
Robert Goodwin, Australia
Silvia Gordillo, Argentina
Feliz Gouveia, Portugal
Virginie Govaere, France
Sven Groppe, Austria
Rune Gustavsson, Sweden
Sissel Guttormsen Schär, Switzerland
Lamia Hadrich Belguith, Tunisia
Thorsten Hampel, Germany
Ajantha Herath, USA
Suvineetha Herath, USA
Francisco Herrera, Spain
Colin Higgins, UK

Peter Higgins, Australia
Tuong Vinh Ho, Vietnam
Erik Hollnagel, Sweden
Jun Hong, UK
Nguyen Hong Quang, Vietnam
Jiankun Hu, Australia
Kaiyin Huang, The Netherlands
Patrick C.K. Hung, Canada
Hamid Jahankhani, UK
Arturo Jaime, Spain
Luis Jiménez Linares, Spain
Luis Joyanes, Spain
Nikos Karacapilidis, Greece
Stamatis Karnouskos, Germany
Hiroyuki Kawano, Japan
Nicolas Kemper Valverde, Mexico
A. Rahman Khan, USA
John Krogstie, Norway
Stan Kurkovsky, USA
Yannick Lallement, Canada
Chul-Hwan Lee, USA
Carlos León de Mora, Spain
Hareton Leung, China
Therese Libourel, France
John Lim, Singapore
Hong Ling, China
Matti Linna, Finland
Jan Ljungberg, Sweden
Stephane Loiseau, France
João Correia Lopes, Portugal
María Dolores Lozano, Spain
Jianguo Lu, Canada
Christopher Lueg, Australia
Edmundo Madeira, Brazil
Laurent Magnin, Canada
Sam Makki, USA
Mirko Malekovic, Croatia
Nuno Mamede, Portugal
João Bosco Mangueira Sobral, Brazil
Esperanza Marcos, Spain
Farhi Marir, UK
Maria João Marques Martins, Portugal
Herve Martin, France
Johannes Mayer, Germany
Andreas Meier, Switzerland
Emilia Mendes, New Zealand
Engelbert Mephu Nguifo, France
Subhas Misra, Canada
Sudip Misra, Canada
Ghodrat Moghadampour, Finland
Paula Morais, Portugal
Fernando Moreira, Portugal
Gianluca Moro, Italy
Haralambos Mouratidis, UK
Ana Neves, Portugal
Jose Angel Olivas, Spain
Peter Oriogun, UK
José R. Paramá, Spain
Vicente Pelechano, Spain
Maria Carmen Penadés Gramaje, Spain
Gabriel Pereira Lopes, Portugal
Laurent Péridy, France
Antonio Pescapé, Italy
Dana Petcu, Romania
Steef Peters, The Netherlands
Paolo Petta, Austria
José Pires, Portugal
Jacek Plodzien, Poland
Geert Poels, Belgium
Macario Polo, Spain
Bhanu Prasad, USA
T. Ramayah, Malaysia
Pedro Ramos, Portugal
Ulrich Reimer, Switzerland
Marinette Revenu, France
Simon Richir, France
António Rito-Silva, Portugal
David Rivreau, France
Daniel Rodriguez, UK
Pilar Rodriguez, Spain
Agostinho Rosa, Portugal
Gustavo Rossi, Argentina
Francisco Ruiz, Spain
Henry Samier, France
Manuel Santos, Portugal
Daniel Schang, France
Arno Scharl, Austria
Mareike Schoop, Germany
Isabel Seruca, Portugal
Hanifa Shah, UK
Jianhua Shao, UK
Charles A. Shoniregun, UK
Alberto Silva, Portugal

Maria João Silva Costa Ferreira, Portugal
Janice Sipior, USA
Hala Skaf-Molli, France
Chantal Soule-Dupuy, France
Chris Stary, Austria
Vijayan Sugumaran, USA
Lily Sun, UK
David Taniar, Australia
Sotirios Terzis, UK
Philippe Thiran, Belgium
Claudine Toffolon, France
Robert Tolksdorf, Germany
Ambrosio Toval, Spain
Theodoros Tzouramanis, Greece
Gulden Uchyigit, UK
Athina Vakali, Greece
Michael Vassilakopoulos, Greece
Christine Verdier, France
Maria-Amparo Vila, Spain
Aurora Vizcaino, Spain
Hans Weghorn, Germany
Gerhard Weiss, Germany
Graham Winstanley, UK
Claus Witfelt, Denmark
Wita Wojtkowski, USA
Robert Wrembel, Poland
Baowen Xu, China
Haiping Xu, USA
Hongji Yang, UK
Kokou Yetongnon, France
Liping Zhao, UK
Shuigeng Zhou, China
Lin ZongKai, China

Auxiliary Reviewers

Lefteris Aggelis, Greece
Muñoz Antonio, Spain
Nick Bassiliades, Greece
Valeria de Castro, Spain
Isabel Nepomuceno Chamorro, Spain
Evandro de Barros Costa, Brazil
Norberto Diaz Diaz, Spain
Samir Elloumi, Tunisia
Christina Evangelou, Greece
Rita Francese, Italy
Vittorio Fuccella, Italy
Frederik Gailly, Belgium
Félix O. García, Spain
Marinos Georgiadis, Cyprus
Carmine Gravino, Italy
Jihong Guan, China
Ignacio García Rodríguez de Guzmán, Spain
Vassiliki Koutsonikola, Greece
Joaquín Lasheras, Spain
Chiraz Latiri, Tunisia
Jose Antonio Gallud Lázaro, Spain
Fernanda Lima, Brazil
Wei Liu, Australia
Stefano Lodi, Italy
Alex Logvynovskiy, UK
Ann Maes, Belgium
Shamila Makki, USA
Leandro Marin, Spain
Sergio di Martino, Italy
Maryvonne Miquel, France
Fernando Molina, Spain
Gabriele Monti, Italy
Nathalie Moreno, Spain
Eduardo Amadeu Dutra Moresi, Brazil
Isabelle Mougenot, France
Joaquín Nicolas, Spain
Germana Menezes da Nóbrega, Brazil
Káthia Marçal de Oliveira, Brazil
George Pallis, Greece
Ignazio Passero, Italy
Hércules Antonio do Prado, Brazil
Wenny Rahayu, Australia
Michele Risi, Italy
Jose-Raul Romero, Spain
Ilias Sakellariou, Greece
Raul Morales Salcedo, Japan
Hanno Schauer, Germany
Manuel Serrano, Spain
Anastasios Sofokleous, UK
Anne Tchounikine, France

Anfana Traore, France
Juan M. Vara, Spain
Belén Vela, Spain
Zhenyu Xie, China
Zhaochun Yu, China
Xiaosu Zhan, China
Lin Zhu, China

Invited Speakers

Georges Gardarin, Université de Versailles St-Quentin en Yvelines, PRiSM, France
Matthias Jarke, RWTH Aachen University, Germany
Timos Sellis, National Technical University of Athens, Greece
John Oommen, Carleton University, Canada

Table of Contents

Invited Papers

Part I – Databases and Information Systems Integration

Part II – Artificial Intelligence and Decision Support Systems

Part IV – Software Agents and Internet Computing

Part V – Human-Computer Interaction

Invited Papers

P2P Semantic Mediation of Web Sources

Georges Gardarin, Florin Dragan, and Laurent Yeh

Laboratoire PRiSM, Versailles University, 45 Ave des Etats-Unis, 78035 Versailles, France
Georges.Gardarin@prism.uvsq.fr, Florin.Dragan@prism.uvsq.fr,
Laurent.Yeh@prism.uvsq.fr

Abstract. XML Mediators are focused on supporting the XQuery (or sometimes the SQL/XML) query language on XML views of heterogeneous data sources. Wrappers leverage data sources to XML views with basic query capabilities. The data are integrated on demand by the mediator delegating sub-queries to wrappers. Using such information integration platforms to query the semantic web is a challenge both for scalability and data semantic reasons. On another side, semantic Peer-to-peer (P2P) networks are emerging as an important infrastructure to manage distributed data, notably on the web. An important goal concerns improving query facilities of distributed heterogeneous data. Coupling data mediation and P2P technology, P2P data mediation strives to support efficiently advanced queries upon heterogeneous data sources annotated with various metadata and mapping schemes. We analyze the main services provided by mediation systems and discuss their extension to the semantic web in P2P mode. We shortly discuss the annotation service for describing sources syntactically and semantically, the routing service to route data location requests, and the query service to resolve distributed queries. We survey some projects and report on PathFinder, an experimental P2P mediation system developed at University of Versailles in the Context of the Satine European project.

Keywords: XML Mediation, XQuery, P2P.

1 Introduction

Semantic heterogeneity is one of the key challenges in integrating and querying data across disparate sources. Systems dealing with heterogeneous data sources have been developed from the early 80's. They were primarily based on the relational model as pivot model and SQL as global query language. With the advent of XML and XQuery, several new EII (Enterprise Information Integrator) have adopted XML as exchange model and XQuery as pivot language. Although these new systems include at their core an XML mediator subcontracting some query processing to powerful XQuery data source adapters, they are insufficient to query the emerging semantic web. Clearly, more distributed search engines and more semantic mapping tools are required to being able to process efficiently thousands of heterogeneous sources.

Peer-to-Peer (P2P) systems are providing attracting solutions to retrieve and share large amount of data among large and dynamic sets of users. The key to scalability is to ship queries only to relevant peers in the overlay network. Unstructured P2P networks are relying on optimizing walks through the network to retrieve relevant data. More

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 3–16, 2008.

convincing are structured P2P networks in which keys are located on peers using distributed hash tables (DHTs). DHTs can be used to maintain fully distributed indexes referring data of a given key on a peer. Thus, search on keys can be efficiently processed by accessing once the DHTs. Keys can be replaced by metadata, thus allowing retrieval of sources handling certain data names or types. Other systems include hybrid networks with special peers, namely super-peers, maintaining indexes refereeing data on peers. Clearly, fully distributed solutions seem preferable for scal-ability and reliability.

Merging XML mediation technology and P2P data source location in large distributed systems seem promising. Relevant servers for processing a query should be efficiently located based on the query information (metadata and content). This does not help in solving semantic heterogeneity. Thus, in addition, several levels of schema with mapping tools relying on common referential (e.g., ontology) should be integrated. Mappings can be performed from local schema to local schema (thus requiring many mappings) or more classically from local views to some global view (GAV, i.e., Global As Views) or reverse (LAV, i..e., Local As Views).

Coupling data mediation and P2P networking, P2P data mediation strives to support efficiently advanced queries upon heterogeneous data sources annotated with various metadata and mapping schemes. The cross-fertilization of the two techno-logies can bring many advantages, among them:

- Efficient discovery of relevant data sources for solving a query.
- New distributed query processing algorithms for complex queries in heterogeneous environment.
- Improved reliability of distributed data management systems.
- Improved scalability of distributed data management systems.

In this paper, we briefly survey the data mediation and P2P network domains. We present our experimental prototypes developed at PRiSM converging towards an efficient P2P mediation system. The rest of this paper is organized as follows: In Section 2, we review the data mediation approach and we present XLive, a mediator developed in the WebSI and Satine European project. Section 3 introduces the P2P network architectures and survey some proposed P2P mediation systems. In Section 4, we describe our PathFinder P2P mediation system built upon Chord DHTs providing XML path discovery for locating data sources to be processed by the XLive mediator. Section 5 concludes the paper and discusses future work.

2 XML Data Mediation

Mediation technology based on XML and XQuery is currently being industrialized. Some products are already available. In this section, we survey this new technology and describe our XLive mediator (see http://xquare.objectweb.org/ for an industrial open source version).

2.1 Basics and Backgrounds

With the advent of XQuery as a standard for querying XML collections [21], several mediator systems have been developed using XQuery and XML schema as pivot language

and model. Examples of full XML mediators are the Enosys XML Integration Platform (EXIP) [16], the Software A.G. EntireX XML Mediator, the Liquid Data mediator of BEA derived from EXIP, and the open source XQuare XML Mediator [23].

XML Mediators are focusing on supporting the XQuery query language on XML views of heterogeneous data sources. Typical architectures (see figure 1) are composed of one or more mediators that deal with distributed XML sources and wrappers that cope with the heterogeneity of the sources (DBMS, Web pages, files, etc.). The data are integrated dynamically from multiple information sources. Queries are used as view definitions. During run-time, the application issues XML queries against the views. Queries and views are translated into some XML algebra and are combined into single algebra query plans. Sub-queries are sent to local wrappers that process them locally and return XML results. Finally, the global query processor evaluates the result, using appropriate integration and reconstruction algorithms.

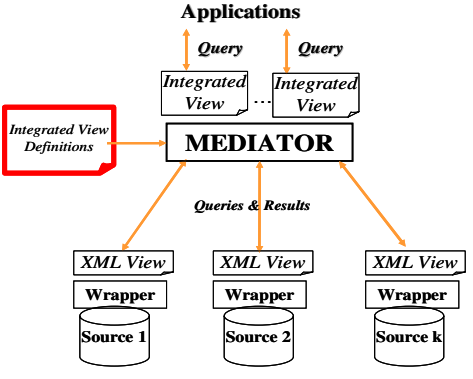


Fig. 1. Typical mediation architecture

XQuery is a powerful language, which encompasses SQL functionalities and much more. XQuery mediators are interesting components to perform semantic integration of Web information. Web mediators aim at transferring Web information from different Web sites in the same domain (e.g., tourism) to a high level domain-specific common information model, making document semantic machine understandable. Web mediators should integrate three kinds of advanced technologies: ontology based description of Web sources, Web data extraction and mapping, and distributed query processing based on mediators.

2.2 Overview of the XLive Mediator

In the XLive project, we use a mediation architecture to support XML information integration. XLive follows the classical wrapper-mediator architecture as explained in [7]. The communication between wrappers and mediator follows a common interface, which is defined by an applicative Java or Web service interface named XML/DBC. With XML/DBC, requests are defined in XQuery and results are returned in XML format.

The XLive mediator is entirely based on W3C standard technology: XML, XQuery, XML-Schema, SAX, DOM and SOAP. To integrate a new source into the mediation architecture, a wrapper must be set up. It has to implement the XML/DBC programming interface, accept some XQuery requests, and return results in XML format. DBMS are data oriented sources and metadata are provided to describe sources and mappings. DBMS wrappers translate data sources in XML and process a possibly reduced set of XQuery on the source data. In the case of Web sources, the wrapper brings more intelligence. It aims at semantically integrating Web information in a common model accessible to programs.

The XLive mediator maintains metadata in a simple way. Each time a source connects to a mediator, it provides the structural tree of the XML documents it is able to retrieve. More precisely, a mediator maintains the set of XML paths that can be queried and retrieved from each wrapper it is connected to. It is called the pathset and can be seen as a weak schema or a summarizing DTD of the source. Query processing uses the path set to check query validity and to determine the relevant data sources for a query. The path set is also used to expand incomplete paths. Pathsets can also be displayed as trees to the final user in order to help him formulate queries. In the sequel, we are going to discuss how to extend these metadata to perform more semantic-based mediation.

The query processing architecture of XLive is represented in Figure 2. XQueries are submitted to the mediator through a graphical interface displaying the pathsets of the integrated views (i.e., the virtual collections). The query is decomposed and passed to each data source according to the previously retrieved metadata. The mediator generates an execution plan, and optimizes the latter to obtain an efficient evaluation. The parameterized execution plan can be saved for other executions. We use the Tree Graph View (TGV) [8] structure to represent queries. The idea is to propose a representation of queries as tree pattern graphs interconnected by hyperlinks. This structure is an extension of the Generalized Tree Pattern graph proposed in [5] as a concise and easy-to-handle repre-sentation of an XQuery More information about the XLive mediator can be found in [8].

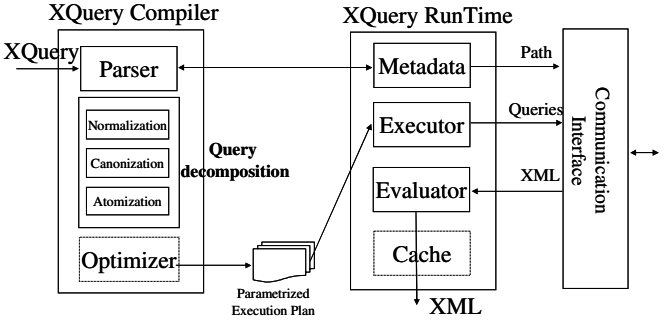


Fig. 2. Query processing in XLive

3 P2P Data Systems

In this section, we briefly review several XML P2P systems that propose efficient architectures for routing complex queries to peers with relevant data.

3.1 Basic P2P Architectures

Classically, P2P networks are categorized in terms of their structures [3]. The literature distinguishes non-structured architectures from structured ones. In distributed non-structured networks, peers are all similar and maintain a registry of their own meta-data. Data discovery is basically done by flooding the network with query requests until relevant sources are located. More sophisticated search in the network may include random walk, local indices and limited radius [6].

Structured networks have very desirable features. They are based on DHTs. With DHTs, objects or references to objects are placed on specific peers determined by hashing a key. Given a key, the corresponding object can be efficiently located using at most O(log(N)) messages, where N is the number of peers. Chord [22] provides a simple and efficient ring structure for the overlay network. It maps both keys and node identifiers onto a virtual ring of hash function values in interval [0, 2^m]. Each key is assigned to its successor node on the ring clockwise (see Figure 3). Nodes and keys are added and removed dynamically, each node being aware of its successor. When a node leaves the network, keys assigned to it are passed to its successor. Queries for a given key are passed around the circle from node to node starting at the query issuing node. Routing tables are maintained (called fingers) to avoid scanning all nodes and dividing the search space by two at each probe. The use of consistent hashing guarantees a good load balancing.

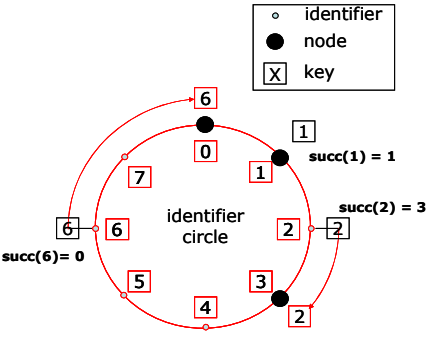


Fig. 3. A Chord identifier ring

Hybrid networks distinguish specific peers (super-peers) from normal peers. Super-peers manage registries of meta-data or indexes giving for each data name or key the address of the relevant peer. Distributed registries or indexes are difficult to maintain when node are often entering and leaving the network. Searching for a name or a key is done by traversing the super-peer network. The super-peer network can be structured in trees or rings to avoid traversing all super-peers.

In the sequel, we survey some P2P systems that consider the structural dimension of data, more sophisticated than the well-known file-name routing techniques (most of them based on DHTs [18], [22]. Mainly, there are two categories of complex routing systems: first, a few papers address the problem of structural routing [17] [19]. Second, there are systems whose routing techniques are based on data semantics [4], [11], [15].

3.2 Structural Routing

In the first category, XPeer [19] is a "self-organizing XML P2P database system" that facilitates storing and querying of XML data. Each peer publishes the hierarchical structure of its shared documents, called a tree-guide. Peers are logically organized into clusters of nodes based on tree-guide proximity. A cluster is indexed by a super-peer that maintains the tree-guides of its peers. Super-peers are organized in a tree. The organization is self-adaptive, which mean that new super-peers can be nominated and peers can be reallocated to clusters when the load of a super-peer is too heavy. A super-peer stores all the schema of its children. Queries are compiled using the super-peer network. This means that the index at super-peers from higher levels is not scalable with the number of peers.

In [17], a P2P architecture is presented for querying XML distributed sources. Queries are routed based on distributed catalogs. Data are localized according to a set of categorization hierarchies similar to dimensions in OLAP and encoded as namespaces. Data providers and consumers use interest areas (i.e., a range in a dimension) to respectively declare data and query data. Index servers and meta-index servers are used to locate interest areas relevant for queries. Category servers maintain the multi-level dimension definitions. Interest areas are encoded in URNs namespace-specific strings. Queries are routed according to URNs matched against meta-index and index. Another originality of the system is mutant query plans, an XML serialization of an algebraic plan that can include in leaves partial results in XML text or URN/URLs. A query is processed by visiting relevant peers and replacing at each node part of the query plan by local XML data. One main assumption of the system is that data distributions and query results can be mapped naturally to multiple hierarchic dimensions.

The MediaPeer system described in [10] is also in the structural routing category. Nodes are organized in peers and super-peers. First level super-peers indexes peers data with an efficient and compact data structure (the trie index) to route queries. Upper layers of super-peers index children super-peers using compact summaries of trie indexes. Query routing is done by matching pathsets involved in queries against the trie index, traversing the hierarchy of super-peers from the root. The system does not suffer from heavy indexes or somehow centralized catalogs to discover data localization. However, root super-peers may become overloaded and mirror super-peers have to be introduced to share load.

3.3 Semantic Routing

In the semantic category, Piazza [11] proposes a P2P infrastructure for sharing and mediating XML and RDF sources. XML peers export XML schemas describing local sources while RDF sources export OWL ontologies. When a peer joins the network, it has to provide mappings between his local schema or ontology and some foreign schema or ontology. Piazza provides support for two methods for semantic mediation: mediated mapping, where data sources are related through a mediated schema or ontology, and point to point mappings, where a local schema is directly mapped to the schema of another site. Mappings are expressed in a language derived from XQuery.

When a user poses a query on a peer, it uses its local schema. The query is first mapped to the local data stored at the node. Next, the routing algorithm determines all

the neighbours of the peer (i.e., nodes related by semantic mappings), reformulates the query for them, and passes the modified queries to them. The recursive processing of a query until no remaining useful links are discovered guarantees the exploration of all sources that are relevant. Sources return partial answers that have to be integrated. In summary, Piazza covers a large variety of mappings with a complex query processing algorithm.

SomeWhere [2] is another project dealing with semantic mediation. It is similar to Piazza, but uses description logic to define mappings between ontologies. Queries are routed according to the relevant mappings.

In P2P data systems, notably for semantic query routing and data integration, metadata are keys. Focusing on metadata standards, Edutella [15] is a multi-staged effort to efficiently handles RDF metadata on top of JXTA, an open source plate-form for supporting P2P applications leaded by Sun. Edutella provides several services, among them a query service and a mapping service, to respectively retrieve metadata and translate between different vocabularies. This infrastructure provides an appropriate routing algorithm for semantic queries based on RDF standards for the description of sources. The routing process uses a network based on a HyperCup [20], which is capable of organizing peers from a P2P network into a recursive graph structure. Edutella should provide an example of routing process based on the semantic of data.

4 Pathfinder P2P Mediator

In this section, we present our new prototype, dealing both with structural and semantic routing on a Chord-based overlay network.

4.1 Objectives and Architecture

The objective of the "PathFinder" project is the integration in the XLive mediator of a P2P level suitable for indexing of XML fragments and efficiently routing of XQueries.

Our previous P2P experiences [10] demonstrate that structure-based clustering of XML paths is a good approach for solving the problem of query routing. As stated above, structure-based routing can be easily implemented in a hybrid P2P network with a hierarchical structure [19], [10]. Comparing to DHT structures, the hierarchical networks present some limitations, mainly regarding the load distribution and the fault tolerance. By adopting efficient reorganization algorithms, the hierarchical systems tend to become efficient but at a high computational cost (the reorganization algorithms require lot of peer computation and message exchanges). To avoid that, we propose a P2P architecture that brings the advantages of the hierarchical networks in a DHT environment. We are mainly interested in adopting a DHT-based architecture where XML paths with similar structure are clustered by proximal peers (DHT clustering).

Due to its guaranteed lookup complexity and easy to understand model, we adopt Chord as the DHT base model. As already presented, Chord assigns identifiers to peers and keys in a specific domain (Chord ring). Each key is stored at the first peer with an identifier that is equal or follows the key identifier. We use the Chord model for structural indexing of XML fragments. For that we assign to each XML path of a

fragment a key in the identifier space and we use the Chord specific method of key indexing.

To cluster XML fragments with similar structure, we propose a mapping technique that assigns to XML paths with similar prefix close identifiers. In this way, with high probability, XML fragments with similar structure are indexed by the same peer or peers close on the Chord identifier ring.

For mapping and indexing each XML path in the P2P network, we use the following steps:

1. Each element of the path is hashed to a numerical value using a hash function keeping the alphabetical order. In this way elements with different structure are mapped to different values and elements with similar structure are mapped to close values.
2. Given the numerical values of all elements, the XML path is mapped to a numerical value using the following map function:

$$Pi = t1/(2\times\alpha+1)^1 + t2/(2\times\alpha+1)^2 + ... + tn/(2\times\alpha+1)^n$$

where t1, t2, …tn represent the numerical values of the XML elements in the path, and α represents the hashing interval (i.e., the maximal value an element can be mapped to).

For illustration consider the following example:

Path1: /catalog/book/author
Path2: /catalog/book/isbn
Path3: /catalog/book/title
Path4: /review/book/title.

Let us assume the following values for the element hashing:

Alphabetical-Hash("author")=1;
Alphabetical-Hash("book")=2;
Alphabetical-Hash("catalog")=3;
Alphabetical-Hash("isbn")=4;
Alphabetical-Hash("title")=5;
Alphabetical-Hash("review")=6.

Using the already presented mapping function, the paths will be mapped to the following values:

Map(Path1)= 3/ 13+2/13^2+1/13^3=0,2429
Map(Path2)= 3/ 13+2/13^2+4/13^3=0,2435
Map(Path3)= 3/ 13+2/13^2+5/13^3=0,2445
Map(Path4)= 6/ 13+2/13^2+5/13^3=0,4517

We observe that Path1, Path2 and Path3, i.e., paths that share a common prefix, are mapped to close values while Path4 is mapped to a distant value. Considering that peers are mapped in the [1/13, 7/13] identifier interval, with high probability (depending on the number of peers) Path1, Path2 and Path3 are indexed by the same peer (or close peers on the identifier ring).

The structural clustering may bias the even distribution of keys. For assuring a good distribution of keys, we manage overflows in case of an overloaded node by

re-indexing pending sub-paths of original paths. Sub-paths in overflow are linked to the prefix of the original paths by pointers.

In our architecture, each peer includes the XLive mediator software and takes part in the P2P network by publishing XML fragments exported by its mediated data sources. The overall system architecture is represented in Figure 4.

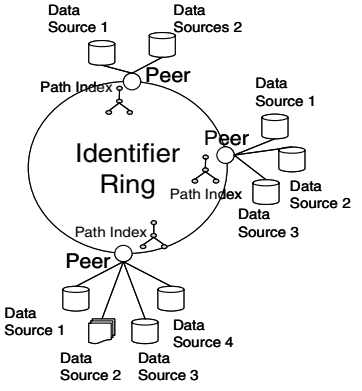


Fig. 4. PathFinder network architecture

4.2 P2P Source Discovery

Data sources are discovered based on the XML fragments published by peers (with mediator structure). After source discovery, the XQueries are sent to peers that are responsible for the publishing of XML fragments. Effective query execution is performed at mediator level: distributed plan generation, plan optimization, plan execution, and result composition.

For discovering data sources, the relevant XML paths are extracted from the submitted XQuery. For each set of paths sharing a common prefix, a DHT-specific lookup is issued. DHT-lookups are routed based on the mappings of the XML paths. The process of source discovery in the P2P network is recursive: the P2P routing starts based on the mapping value of the common prefix of each set of paths and continues searching the remaining sub-paths.

We base our source discovery algorithm on the following observation: given an XML path, the mapping value of one prefix of the path is less than the mapping value of the whole path. For example, consider one of the paths presented in the previous example:

Path1: /catalog/book/isbn

The following inequality is enforced:

$$Map(/catalog) < Map(/catalog/book) < Map(/catalog/book/isbn)$$

This means that the peer indexing "/catalog" (if a path "catalog" was published) is situated before the peer indexing "/catalog/book" on the identifier ring, then, at its turn, situated before the peer indexing "/catalog/book/isbn".

If the path "/catalog" is not indexed, the potentially lookups for the "/catalog" prefix are routed to the peer indexing paths starting with "/catalog". Moreover, the last two peers take part in the set of fingers of the first peer (considering the full Chord architecture with fingers as routing tables). This means that forwarding a query from the first peer to the next requires only a new step.

By adopting a source discovery routing algorithm based on the discovery of common path prefix first, the process is optimized: a common prefix is routed a single time.

4.3 Query Execution

Once the sources are discovered, the atomic XQueries (queries that refer to a single data source) are executed by the peer that published relevant paths (running the local wrapper), the local results are sent to the client peer, and the final result is composed by its mediator. The client mediator basic task is the generation of an "optimal" logical execution plan, the search for the sources, and the result composition by the execution of the global part of the plan.

For illustration, we consider the following XQuery:

```
for $b in collection("catalog")/book
for $r in collection("review")
where $b/author="Alon"
and $b/title=$r/book/title
and $r/book/title contains("Query processing")
return $b/isbn
```

At the P2P source discovery level the following two pathsets are extracted:

Set1: /catalog/book/author, /catalog/book/title, /catalog/book/isbn
Set2: /review/book/title

The peer(s) where one or the two sets are routed execute one or the two following atomic queries upon its wrapped data sources:

```
for $b in collection("catalog")/book
where $b/author="Alon"
return $b/title,$b/isbn
and
for $r in collection("review")
where $r/book/title contains ("Query processing")
return $r/book/title
```

The final union and join operations are executed by the mediator at the peer that initiated the query. For being able to execute an atomic query a single data source must satisfy all the paths from the atomic query. For combining parts of different data sources, the mediator view engine can be used. More information regarding the query execution at mediator level and view processing can be found in [7], [12].

4.4 XML TE

In this sub-section we present another P2P prototype used for publication and querying of XML data. Comparing to PathFinder this prototype is not based on a lower

level DHT. On the contrary it is based on a P2P indexing platform that is optimized for XML data.
The architecture is composed of two layers:

- an overlay layer formed of physical components named hubs denoted by "hub-network".
- an overlay layer composed of the actual peers organized according to the principles of Skip-Graphs and Skip-Lists.

In figure 5 is presented the general two-layered network.

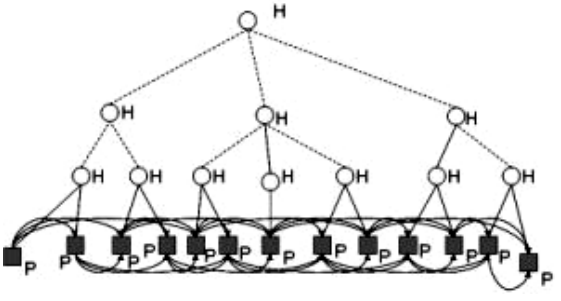


Fig. 5. Generic Network Architecture

The Hub-Overlay: The main role of the hub overlay is to organize the data load distribution to the peers in the peer overlay. It's structure depends on the schema of the indexed XML data (figure 6).

The Peer-Overlay: The peer overlay is formed of all peers that take part in the P2P network. Each peer stored data chunks assigned by the upper hub-overlay.

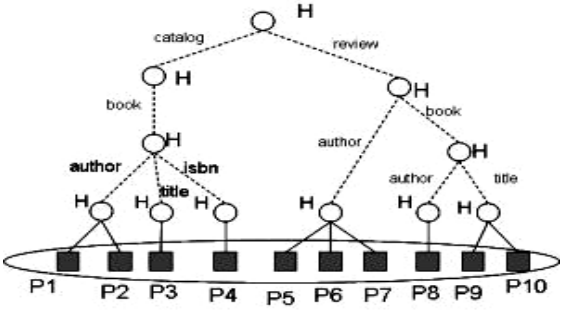


Fig. 6. Network Instance

Query Execution: As in PathFinder the architecture is adapted to execution of simple path queries and complex XML queries (e.g. range queries). Each XQuery is decomposed in a set of XPaths. Each path is routed from peer to peer in the second network layer. As data is well organized the routing decisions are easily computed.

For example given the path *"/review/author*" at P1 and considering that P1 "knows" the values indexed by P2, P3, P4 , P1 will directly forward the query to P4 that, based on local index information (P5, P6, P8) will deliver it to P5 (first connection).

Similarly prefix queries (or range queries) can be easily solved. For example consider that P1 searches for all peers that contain data with the path prefix "/catalog/ book". Based on the local index information (e.g. skip-lists) the query is forwarded to all peers between P1 and P4. We remember that this routing principle can be applied because of the ordering of the indexed data.

Comparing to the PathFinder architecture in XML TE no path re-indexing is triggered by a heavy network load. Currently in the XML TE architecture is developed a semantical layer that helps in routing queries to the peers that contain relevant data by taking into consideration the semantic descriptions of data.

4.5 The Semantic Layer

Integrating XML data at the structure level is meaningless in the case of fully heterogeneous sources, in which tag names and tag nesting have different meanings. For example book/person in XML may mean book written by a person in one source and book bought by a person in another. Person/name can be encoded simply as person in another source. Price can be in different units, and so forth. Thus, a semantic layer has to be provided to perform uniform and non-ambiguous integration of data sources.

In PathFinder, we opted for mediated mappings, where data sources are related through a domain ontology. In addition to be XML/XQuery wrapped, each source has to be semantically wrapped. The semantic wrapper maps the local pathsets to semantic pathsets that are derived from the mediated ontology. The ontology is expressed in a subset of OWL. A graphical interface allows the user to see the domain ontology and its local XML view. Using drag and drop, each path of the local view has to be mapped to a path included in the domain ontology, which gives a path with a domain meaning. As a result, the semantic wrapper yields a set of statement according to a domain ontology. For example, *book/person* can be mapped to *book (is bought by) author*. Our system indexes these RDF statements in the P2P network according to the above described mechanisms.

For posing queries, the user refers XML views and uses XQuery. Each view is a virtual collection with a pathset. The paths have to be mapped non-ambiguously to a corresponding path within the domain ontology. Each view corresponds to a hierarchically arranged subset of the global ontology. This subset gives the sub-domain meaning expected in the data sources to be retrieved on the network. The path mapping should be defined using a graphical tool.

Using the semantic paths, relevant sources can be looked up in the "semantic" DHT as explain for logical queries. When relevant sources have been retrieved, relevant data should be accessed, transformed, and shipped back to the requestor for final integration. We currently explore two ways for mapping the data:

1. Ad hoc mapping, with path to path correspondence, derived through the "same as" relationship between paths. This can be extended with functions applied at certain data elements (e.g., to translate euros in dollars). This approach is simple, can be efficiently implemented, but is limited in mapping capabilities.

2. Using XQuery to define the mapping from the local view to the "ontological" one, and also from the ontological views to the user ones. XQueries should then be generated by the mapping tools. This approach has some similarities with the Piazza one in case of mediated mappings. It is more complex but probably more powerful than path to path mapping.

5 Conclusion

In this paper, we first survey the data mediation technology and illustrate it with XLive, an XML mediator supporting XQuery on XML views of various data sources. Then, we focus on P2P networking that promises to leverage existing mediation technology to manage large numbers of heterogeneous sources. We survey several key research prototypes before presenting in some details our PathFinder system being developed partly in the context of the Satine project.

Satine is a more general European project [9] using a hybrid P2P architecture, with peers and super-peers. In the future, we intend to compare the PathFinder DHT-based architecture with the Satine hybrid one. Path order-preserving hashing functions allowing range query and balanced hashing as integrated in PathFinder should provide good scalability characteristics.

References

1. Abiteboul, S., Manolescu, I., Preda, N.: KadoP: Knowledge and Data in Peer to Peer (2004), http://www-rocq.inria.fr/gemo/KadoP/
2. Adjiman, P., Chatalic, P., Goasdoue, F., Rousset, M.-C., Simon, L.: SomeWhere in the Semantic Web. In: Fages, F., Soliman, S. (eds.) PPSWR 2005. LNCS, vol. 3703, Springer, Heidelberg (2005)
3. Androutsellis-Theotokis, S., Spinellis, D.: A survey of peer-to-peer content distribution technologies. Computer Surveys 36(4), 335–371 (2004)
4. Bernstein, P., Giunchiglia, F., Kementsietsidis, T., Mylopoulos, J., Serafini, L., Zaihrayeu, I.: Data management for peer-to-peer computing: A vision. In: Proceedings of the Fifth International Workshop on the Web and Databases, Madison, Wisconsin (2002)
5. Chen, Z., Jagadish, H., Lakshmanan, L., Paparizos, S.: From Tree Patterns to Generalized Tree Patterns: On Efficient Evaluation of XQuery. In: Proceedings of the 29th International Conference on Very Large Data Bases, Berlin, Germany (2003)
6. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: Proceedings International Conference on Distributed Computing Systems (2002)
7. Dang Ngoc, T., Gardarin, G., Jamard, N., Travers, N.: Federating Heterogeneous Data Sources with XML, IASTED IKS 2003, Scottsdale, AZ, USA (2004)
8. Dang Ngoc, T., Gardarin, G., Travers, N.: Tree Graph View: On Efficient Evaluation of XQuery in an XML Mediator. In: BDA. Proc of 20th conf. Bases de Données Avancées, Montpellier, France (2004)
9. Dogac, A., et al.: The Satine Project: A P2P Network for Discovering Web Services (2004), http://www.srdc.metu.edu.tr/webpage/projects/satine/
10. Dragan, F., Gardarin, G., Yeh, L.: MediaPeer: a Safe, Scalable P2P Architecture for XML Query Processing. In: Globe 2005. Workshop Copenhagen, Denmark (2005)

11. Halevy, A., Zachary, I., Mork, P., Tatarinov, I.: Piazza: Data management infrastructure for semantic web applications. In: WWW 2003. Proceedings of the Twelfth International World Wide Web Conference, Budapest, Hungary (2003)
12. Jamard, C., Gardarin, G.: Indexation de vues virtuelles dans un médiateur XML pour le traitement de XQuery Text. EGC 2006, Lille (2006)
13. Löser, A., Siberski, W., Wolpers, M., Nejdl, W.: Information integration in schema-based peer-to-peer networks. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, Springer, Heidelberg (2003)
14. Morrison, R.: PATRICIA - Practical Algorithm to Retrieve Information Coded in Alphanumeric. Journal of the Association of Computing Machinary 15(4), 514–534 (1968)
15. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I.: Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks. In: Proceedings of the International World Wide Web Conference, Budapest, Hungary (2003)
16. Papakonstantinou, Y., et al.: Queries and algebra in the Enosys Integration Platform. Data & Knowledge Engineering 44(3), 299–322 (2003) (Special issue: Data integration over the Web)
17. Papadimos, V., Maier, D., Tufte, K.: Distributed Query Processing and Catalogs for Peer-to-Peer Systems. In: CIDR. Conference on Innovative Data Systems Research, Asilomar, USA (2003)
18. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. In: Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications, New York, USA (2001)
19. Sartiani, C., Manghi, P., Ghelli, G., Conforti, G.: XPeer: A Self-organizing XML P2P Database System. In: P2P&DB 2004. Proceedings of the First EDBT Workshop on P2P and Databases, Crete, Greece (2003)
20. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: HyperCuP-Hypercubes, Ontologies and Efficient Search on P2P Networks. In: International Workshop on Agents and Peer-to-Peer Computing, Bologna, Italy (2002)
21. Scott, B., et al.: XQuery 1.0: An XML Query Language (2005), http://www.w3.org/TR/xquery/
22. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications, New York, USA (2001)
23. XQuare. The XQuare project: open source information integration components based on XML and XQuery (2005), http://xquare.objectweb.org/

Reflective Community Information Systems

Matthias Jarke and Ralf Klamma

Information Systems, RWTH Aachen, Ahornstr. 55, 52074 Aachen, Germany
jarke@cs.rwth-aachen.de, klamma@cs.rwth-aachen.de

Abstract. The Internet has not only enabled worldwide access to heterogeneous information sources such as web pages or traditional database contents, but also increasingly serves as a medium for multimedia information and opinion exchange. Community Information Systems address the combination of these two trends of heterogeneous worldwide information access and cooperative discussion and work. This combination creates a lot of new opportunities e.g. in the educational and cultural sector, but entails also serious risks and socio-political problems. New technical solutions are required for problems such as share definition of IS structure in such communities, high variability and strong guidance in user interfaces, security and trust management. In particular, this requires a schema organization that can adapt itself gradually, yet in a controlled manner, i.e. has the property of being reflexive. This paper gives an overview of interdisciplinary research in community information systems, and presents the ATLAS architecture developed at RWTH Aachen University. A number of real-world applications, including a major reconstruction effort for a cultural heritage research community in Afghanistan, illustrate the approach.

Keywords: Social software, information systems engineering, cooperative information systems.

1 Internet Communities and Social Software

Starting with open Internet-based groupware tools such as the BSCW [1], the world wide web has evolved from a publication medium to a cooperation medium. *Communities of practice*, i.e. self-organized social groupings with shared interests or activities [19], include the Internet among their preferred environments for mutual learning and reflection. Internet-based communities often span geographical as well as organizational boundaries. In times of less and less stable institutions, they become an important source of social identity, but they can also constitute an intra- or inter-organizational memory.

Internet communities produce their own cultures which begin to play a significant role also in the political sector. However, they are prone to disturbances and very hard to predict or control. Relatively harmless examples are the clumsy attempts of PR departments within large companies to post positive presentations of their top managers in Internet lexica such as Wikipedia, or to remove critical ones. More serious is the empirically well-supported observation that Internet communities may contribute significantly

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 17–28, 2008.

to the prolongation and brualization of the so-called asymmetric wars and civil wars we are faced with since the beginning of the new millenium [12].

Internet communities are an important bottom-up complement to top-down process modeling that has dominated the enterprise information systems discussion of the last decade and have expressed themselves in process modeling tools such as ARIS [14] and ERP environments such as SAP R/3 or Oracle Financials. This implies firstly the research question of mechanisms for the analysis, organization and reflection within Internet communities analogous to process modeling and process management.

Secondly, the question arises how the many barely interoperable media and artefacts which are meanwhile being used by communities, can be integrated in such analyses and environments.

Table 1. Heterogeity of media and artfacts in Social Software

Artefacts Medium	Email	News-letter	Discussion Group	Blog	Transactional Web Site	Wiki	Chat Room	URL
Message	+	+	+	-	-	-	-	-
Burst	+	-	+	+	+	+	-	-
Blog Entry	-	-	-	+	-	-	-	-
Thread	+	-	+	-	-	+	+	-
Web Page	-	-	-	-	-	+	-	+
Comment	-	-	-	+	+	+	-	-
Transaction	-	-	-	-	+	-	-	-
Conversation	-	-	-	-	-	+	-	-
Feedback	-	-	-	-	+	-	-	-

Table 1 juxtaposes the most important types of the continuously growing flood of media and artefacts, fostered in particular by the rapidly growing usage of multimedia elements such as speech, photos, video, and advanced graphics up to virtual and augmented reality. Leading software providers currently attempt to structure this variety in so-called Collaborative Computing environments, but this is approved only by part of the communities.

In its entirety this zoo of media and artefacts is currently debated under the label of *social software* [17]. According to this discussion, social software should facilitate the *construction* of social networks among individuals and groups, as well as *conversation-based interaction* in these networks independent of specific technologies or communication channels. In contrast to traditional groupware the focus is placed on voluntary participation and on self organization with rapidly changing roles between consumers and producers of information. Social feedback leads to an emphasis on emerging rather than pre-defined structures in which reputation builds less on formal criteria than on stepwide building of trust among network partners.

In the sequel, we first discuss the linkage between social software and process management, in particular the role of metadata. Then, we present a reflective software Architecture for Transcription, Localisation, and Addressing Systems (ATLAS)

which aims at supporting community information systems, and sketch our prototypical implementation of this architecture. In ATLAS, reflection is supported by novel pattern-based analysis methods for social networks which we have developed as an extension of our earlier work in organizational requirements engineering. Finally, we report experiences with a few ATLAS applications and give an outlook at further research needs.

2 ATLAS: Metadata and Reflective Information Systems

Social software the currently best-known example of which are probably the web logs (briefly: blogs) has also begun to play a serious role in many enterprises, as an instrument of knowledge management, of project management, or of public relations. Under the buzzword of Web 2.0 the Internet becomes a "network of interaction". Important contributors to this success are low entry barriers, easy usability, and great flexibility.

In contrast, the structuring of these information networks, e.g. in order to interface them with formal process management, is far more difficult. Bottom-up "community of practice" and top-down "business process management" (supported by ERP) are fundamentally different perceptions of labor. A productive reconciliation of these perspectives requires clever metadata management, as indicated in fig. 1, an adaptation of the well-known industrial engineering or labor science triangle of people, organization, and technology [18].

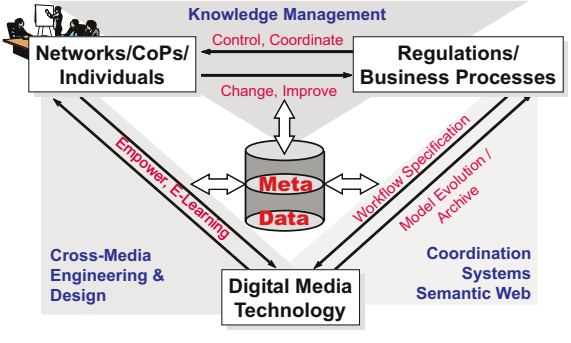


Fig. 1. Metadata link social software and processes

An architecture intended to support community information systems on this basis requires particular flexibility. On the one hand, arbitrary multimedia artefacts with different development histories, semantic meaning, device properties and usage rights need to be integrated such that users with widely varying competence and interest profiles, but also with different conceptual perceptions on content can work with them. On the other hand, especially a community in an enterprise setting needs to be able to observe itself, to analyse and maybe even simulate its behavior, in order to evolve its rules of cooperation, if it wants to survive the commonly occuring disturbances.

The above considerations originate from an operational theory of media called *Transcriptivity Theory* [5]. Transcriptivity Theory was developed in the Collaborative Cultural Science Research "Media and Cultural Communications" (SFB 427) operated by the Universities of Cologne and Bonn jointly with RWTH Aachen University. It describes the operational semantics of medial artefacts based on three ground operations:

- Intra-media or inter-media *transcription* structures a collection of existing media objects and interprets it through a new media object.
- *Addressing* specific target groups through the new media object takes up the competition for attention within these target groups.
- In turn, members of the target groups have their own strategies for *localization* of arriving information, i.e. their own attention management and interpretation framework for arriving media objects.

Due to this theoretical underpinning, we call the software architecture in which we have summarized the above requirements ATLAS (Architecture for Transcription, Localization, and Addressing Systems), see fig. 2.

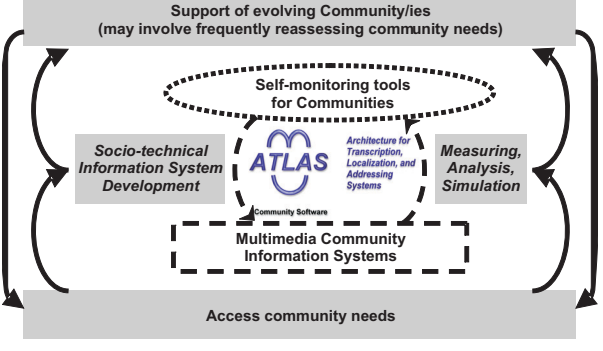


Fig. 2. ATLAS architecture for reflective community information systems

The ATLAS architecture has been mapped into a software framework and was implemented in a stable prototype. The ATLAS prototype *administers the metadata* in structured repositories and *offers the related services* via an extensible webservice provider, as described below.

Scalable and interoperable repositories on top of XML databases support ATLAS communities by technologies for the creation, maintenance, and usage of metadata. Metadata structures follow international multimedia standards such as MPEG-21 (see fig. 3) which can document structures (e.g. collections, schemas, links) and views on them, as well as process aspects (creation, real world denotation, usage rights), user preferences and usage histories. In fig. 3, the small subfigure to the lower right illustrates some of the metadata objects using photographs of the famous Buddha statues in Bamian/Afghanistan before and after the destruction by the Taliban in early 2001.

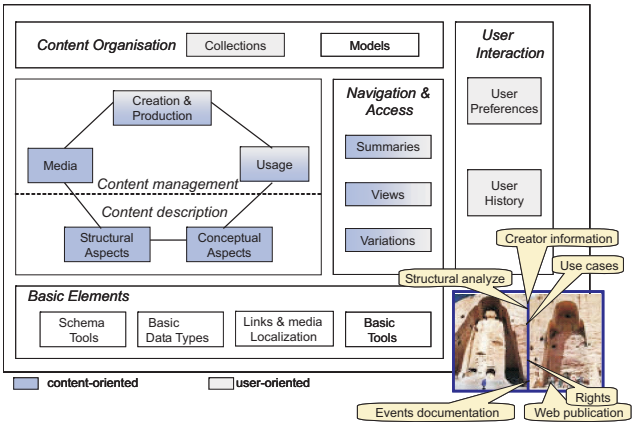


Fig. 3. MPEG-21 multimedia metadata structure

Services of the metadata repositories are offered as web services through a *Light-Weight Application Server (LAS)* we have developed (fig. 4).

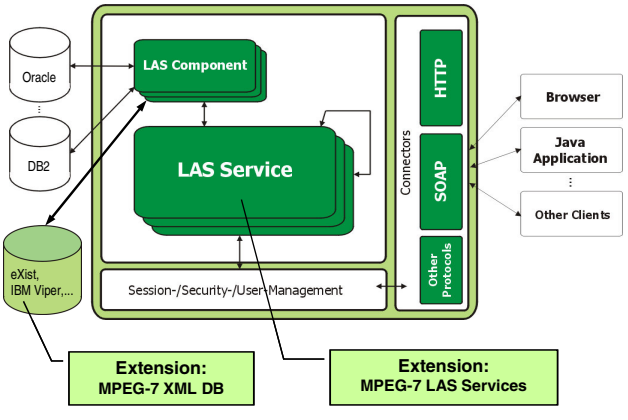


Fig. 4. ATLAS Extensible Lightweight Application Server

To achieve the property of reflection, it was decisive that the service offerings can be extended at runtime. Services can also be hierarchically organized in the sense that high-level services can be composed from relatively simple basic services.

As an example of a higher-level service, fig. 5 shows a semantic image tagger. Besides the functionality of semantically annotating images using keywords as well as entries from concept and process ontologies (upper part), it also feeds automatically an interactive graphical network analysis (lower part) that links the tagged media artefact to others in the repository.

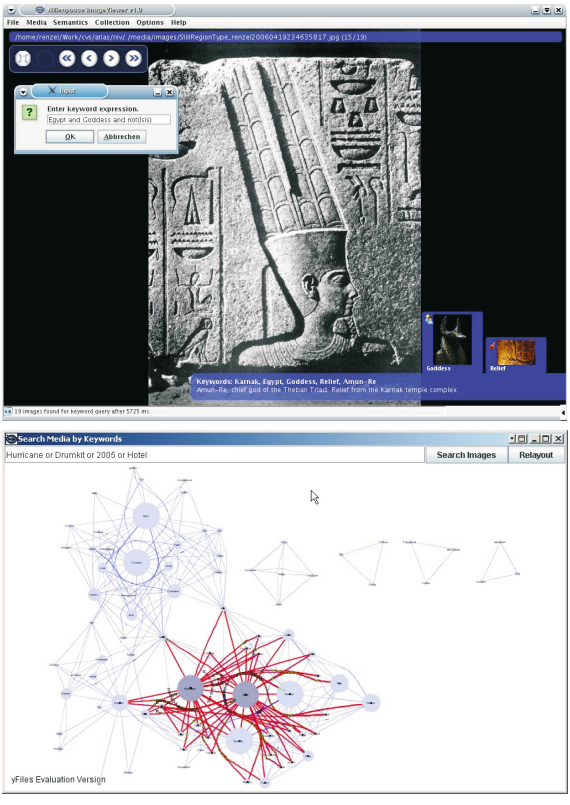


Fig. 5. Semantic image tagging as a higher-level composite web service in ATLAS: Nodes in the concept graph within the lower part of the figure correspond to the concepts by which images are annotated, and thus provide a semantic linkage between media artefacts

3 Tools for Modeling and Analysis

As ERP applications require process modeling tools for analysis and design [14], we believe that communities also need tools for analyzing their interactions. However, in settings where orderly processes are initially not even existent, the question is not just *what* modeling formalism might be suitable here but even *if* process modeling makes any sense at all. Indeed, a long tradition of verbal descriptive ethnological analysis in social networks [2] is rather hostile towards formal modeling. However, in designing ATLAS, we pursue the hypothesis that for the runtime self-analysis of disturbances within internet communities, we do not only need a certain degree of formality but even largely automated model analysis, exactly because community members have little interest or competence in spending time in data analysis.

Our modeling framework builds on earlier requirements engineering work using agent-based modeling of enterprise cooperations under special consideration of trust

modeling. Our TCD (Trust, Confidence, Distrust) model [4] is a multi-perspective requirements engineering approach [11], [6] that initially comprised four partial models.

At the *strategic level*, the i* formalism [21] represents the strategic goal hierarchy of individual network members as well as the strategic dependencies among them. At the *operational level* goals are mapped to process definitions, whereas speech-act-based workflow models of task delegation implement the strategic dependencies. Speech-act based workflow models, initially proposed by [20], have been successfully used especially in service and eBusiness settings [13], [15]. [4] derived a dynamic trust model from tracing the workflow reality against the delegation agreements and goals.

Nevertheless, the TCD model is still based on the assumption of ex-ante manual construction of the partial models and their relationships. This would certainly exceed the capabilities and willingness of typical internet community organizers, as leadership and coordinator roles tend to shift frequently within communities. Another reason is simply that our understanding of community processes is still in its infancy. Furthermode, the TCD model does not consider the crucial role of using multiple media and artefacts which we have identified as a growing characteristic of many internet communities.

We have therefore combined some of the basic TCD modeling ideas with Latour's (1999) *Actor-Network Theory (ANT)* and with community interaction patterns, a concept we have developed analogous to software patterns [3].

ANT is a theory of social networks from the mid-1990's which for the first time did not distiguish any more between human and technical actors. Interaction semantics between actors or actor groups (called *members* of the network) is defined by the context of digital social networks in which they operate. Basic categories are the available *media* that provide members the infrastructure to exchange information, and specific *artefacts* that are exchanged by these media or media combinations.

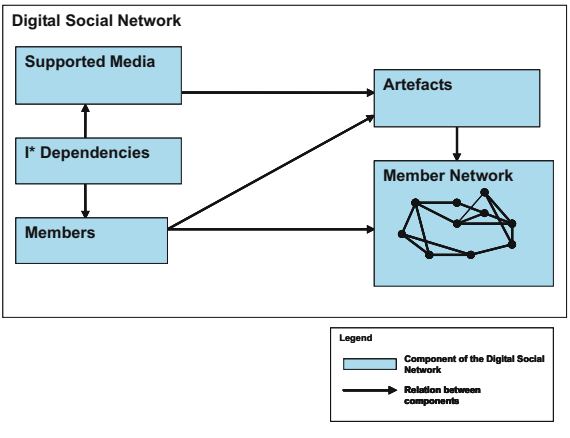


Fig. 6. Simplified media-aware Actor-Network Metamodel as a basis for analyzing digital social communities

In comparison to TCD we can say that ANT shifts the semantics of the intention in communication offered, e.g., by speech-act theory as the basis for workflow modeling, towards the usable media and the exchanged artefacts. This shift also defines the

space for different kinds of social software, as already indicated in table 1. Taken together, we obtain a *media- and artefact-aware social network metadata model* as sketched in fig. 6. This metamodel overlaps in an obvious manner with the ATLAS repository structure discussed earlier. It can thus serve as the basis for interaction data analysis within the ATLAS framework.

In a first refinement of the very rough meta model of fig. 6, we can easily subclassify special member types which are well-known from the social network analysis literature [16], such as hubs (nodes with many connections) or gatekeepers (bottleneck nodes through which most or all connections between two subnets pass). We can also formally characterize certain disturbance patterns in networks such as spammers (who flood their environment with undesired multiple copies of the same artefacts) or trolls (who tend to hold conversations only with themselves).

The automatic recognition of such patterns is supported by PALADIN [7], an interaction pattern analysis system based on our meta model. PALADIN extends the well-known methods for the analysis of scale-free networks [16] by media aspects. A logic-based constraint language for the definition of (desirable or undesirable) interaction patterns in communities is the basis for searching specific instances of such patterns on the interaction graph of the community.

Each pattern comes annotated with user documentation: argumentative explanations why occurences of this pattern could be useful respectively problematic for the community, and experience-based hints for members and coordinators of the community how to deal with such patterns.

Technically, PALADIN operates on the XML structures of the ATLAS repository and is offered to the users via a web interface through the LAS. The system has been calibrated successfully with large communities of up to several thousand members and shows acceptable performance. The next step in our research are experiments in which we want to evaluate if communities can actually use the recommendations associated with the patterns and what actual impact these recommendations have on success criteria of the communities.

4 Application Experiences

A number of practical ATLAS-based applications have been developed in cooperation with colleagues from user community disciplines. Most of them are located in the field of technology-assisted learning.

The MECCA environment [8] supports communities of teachers, researchers, and students in the movie sciences in the cooperative viewing and semantic annotation of large collections of movies. The different formal and content-related conceptualizations pursued by competing schools of thought can be visualized explicitly and thus becomes subject of fruitful scientific and educational debates.

The social network analysis tools offered by ATLAS were first applied in the context of studying the evolution of interdisciplinary and international cooperation structures in large projects such as SFB 427 or the European Network of Professional Elearning, PROLEARN [7].

The *Afghan Cultural heritage Information System ACIS* [9] shown in fig. 7 pursues a significantly more ambitious goal.

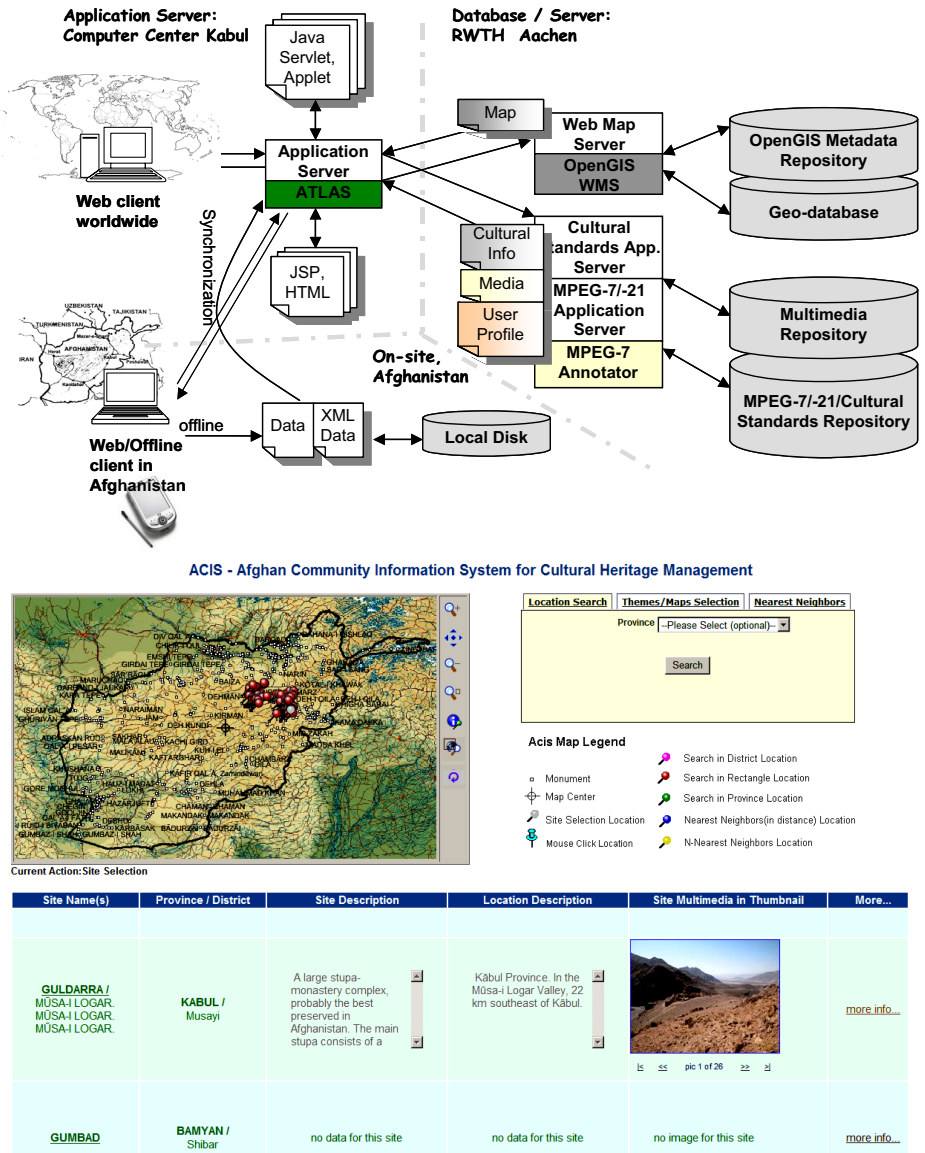


Fig. 7. ACIS architecture and screenshot

More than two decades of external and civil war in Afghanistan have interrupted research in the rich cultural heritage of the country for a full generation from before 1980 up to well beyond 2001.

Researchers of the old generation, if they survived at all, are distributed over the whole world and have developed a deep and often well-justified distrust against

contributing their preserved knowledge and media artefacts. At the same time, the young local scientist generation that has restarted this scientifically as touristically highly important field in the last few years, have basically no access to earlier research documents and knowledge. Thus, we have a problem of inter-generational as well as geographic diversity.

In cooperation with the Urban Historian Michael Jansen – one of the first international scientists to visit key cultural heritage sites after the toppling of the Taliban in late 2001 – and supported by several international organizations, ACIS has been developed on top of ATLAS as a multimedia geographic community information system intended to facilitate the trusted cooperation and knowledge exchange between the grandparents in the diaspora and grandchildren in Afghanistan itself.

Key design aspects of ACIS were firstly a solid database infrastructure with particular emphasis on security and portability. A major *security* threat is e.g. that cultural heritage re-discovered by community cooperation, falls victim to terrorist attacks or international art theft. *Portability* is important because a gradual handover of the database server functions and contents should be possible from the initial German site which was chosen for testing, IT competence, and security reasons, to local ownership and operation.

A second major challenge was the combination of multimedial and spatio-temporal information structures which is typical for this application domain. This challenge includes the integration of mobile personalized data capture and display capabilities in a local setting where still only weak communications and power infrastructure exists.

The main design of communication patterns in ACIS is the stepwise creation of mutual trust. This implied the design of certain security-related interaction patterns but also of reliable documentation strategies which help to slowly reduce the distrust that has been built up in decades, despite a situation where network members have little chance to meet face to face.

This process is far from complete today, as new threats are emerging in the local arena. Nevertheless, the system could be developed to a degree of social and technical networking, and to a quantity and quality of content that the Afghan government has decided to embed it in its reconstruction strategy for Afghan cities. This strategy wants to preserve and promote cultural heritage despite the difficult financial and security situation in the country. The first culture-aware town reconstruction plan has meanwhile been approved by both the central government and the local decision makers. This will also be a first example of interfacing our more informal community information system approach with ERP issues such as financial control and project planning. The longer-term vision is an e-governance approach which combines legal clarity and accountability with citizen-oriented communities.

5 Summary and Outlook

In this paper, we have presented an attempt to transfer the successful idea of business process modeling and analysis from the usual top-down management perspective, to the setting of large, bottom-up self-organizing communities operating through multiple internet-based media.

The initial results show that a number of basic ideas appear transferable but the modeling formalisms and analysis tools required are rather different. Nevertheless, there are a few shared concepts, of course mainly that of an organizational actor, which inspire hope that top-down and bottom-up – more practically speaking ERP and Social Software – might be reconciled in the future both technologically and conceptually.

Acknowledgements

This work was supported in part by the Deutsche Forschungsgemeinschaft within Collaborative Cultural Science Research Center "Media and Cultural Communicatons" (SFB 427), by the European Union within the PROLEARN European Network of Excellence on Professional Technology-Enhanced Learning, and by the Finnish Crisis Management Initiative CMI. We are grateful to our colleagues Yiawi Cao and Marc Spaniol for their technical contributions, and to our application partners for their significant design suggestions and feedback in the evaluation of this work.

References

1. Bentley, A., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkel, S., Trevor, J., Woetzel, G.: Basic support for cooperative work on the world wide web. Human-Computer Studies 46(6), 827–846 (1997)
2. Degenne, A., Frose, M.: Introducing Social Networks. Sage Publications, Thousand Oaks (1991)
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1994)
4. Gans, G., Jarke, M., Lakemeyer, G., Schmitz, D.: Continuous requirements management for organization networks: a (dis-)trust-based approach. Requirements Engineering 8(1), 4–22 (2003)
5. Jäger, L.: Transkriptivität: Zur medialen Logik der kulturellen Semantik. In: Jäger L/Stanitzek, G. (ed.) Transkribieren - Medien/Lektüre, pp. 19–41. Fink-Verlag, München (2002)
6. Kethers, S.: Multi-perspective modelling and analysis of cooperation processes. Dissertation, RWTH Aachen University, Germany (2000)
7. Klamma, R., Spaniol, M., Cao, Y., Jarke, M.: Pattern-based cross-media social network analysis for technology-enhanced learning in Europe. In: Nejdl, W., Tochtermann, K. (eds.) EC-TEL 2006. LNCS, vol. 4227, Springer, Heidelberg (2006)
8. Klamma, R., Spaniol, M., Jarke, M.: MECCA: Multimedia capturing of collaborative scientific discourses about movies. Informing Science 8(1), 3–38 (2005a)
9. Klamma, R., Spaniol, M., Jarke, M., Cao, Y., Jansen, M., Toubekis, G.: ACIS: Intergenerational community learning supported by a hypermedia sites and monuments database. In: Proc. ICALT, Kaohsiung/Taiwan, pp. 106–112 (2005b)
10. Latour, B.: On recalling ANT. In: Law, J., Hassard, J. (eds.) Actor-Network Theory and After, pp. 15–25. Oxford Univ. Press, Oxford (1999)
11. Nissen, H.W., Jeusfeld, M.A., Jarke, M., Huber, H.: Managing multiple requirements perspectives with meta models. IEEE Software 13(2), 37–48 (1996)
12. Rao, A., Bollig, M., Böck, M.: Practise of War. Berghahn Books, Oxford (2006)

13. Schäl, T.: Workflow Management Systems for Process Organisations. LNCS, vol. 1096. Springer, Heidelberg (1996)
14. Scheer, A.-W.: ARIS: Vom Geschäftsprozess zum Anwendungssystem. Springer, Heidelberg (1998)
15. Schoop, M., Quix, C.: DOC.COM: a framework for effective negotiation support in electronic marketplaces. Computer Networks 37(2), 153–170 (2001)
16. Scott, J.: Social Network Analysis - A Handbook, 2nd edn. Sage Publications, Thousand Oaks (2000)
17. Shirky, C.: Social software and the politics of groups. Networks, Economics, and Culture (2003), \url{http://www.shirky.com/writings/group_politics.html}
18. Ulich, E.: Arbeitspsychologie. Schäffer-Pöschel, Stuttgart (1992)
19. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press, Cambridge (1998)
20. Winograd, F., Flores, F.: Understanding Computers and Cognition: A Foundation for Design. Ablex Publ., Greenwich (1986)
21. Yu, E.: Modeling Strategic Relationships for Process Reengineering. Ph.D. Thesis, University of Toronto, Canada. Technical Report DKBS-TR-94-6 (1995)

Data Exchange Issues in Peer-to-Peer Database Systems

Verena Kantere and Timos Sellis

Dept. of Electrical and Computer Eng., Computer Science Division
National Technical University of Athens, Athens, Greece
verena@dblab.ece.ntua.gr, timos@dblab.ece.ntua.gr

Abstract. Sharing of structured data in decentralized environments in the absence of a global schema is a challenging problem. In the context of P2P database systems we are investigating methods for the better fulfilment of peer-database information needs. Towards this end, we focus on the problem of query degradation along the propagation path. The solution we develop is a method that utilizes only normal query traffic in order to discover remote veiled peers with interesting information. Furthermore we intend to work on the dynamic exploitation of local knowledge for the general benefit of the P2P system.

Keywords: Peer-to-peer systems, query processing.

1 Introduction

The P2P paradigm dictates a fully distributed, cooperative network design, where nodes collectively form a system without any supervision. In contrast to data integration architectures, P2P data sharing systems do not assume a mediated schema to which all sources of the system should conform in order to share data. In such a system, where peers share structured data, each peer is an autonomous source that has a local schema. Sources store and manage their data locally, revealing only part of their schemas to the rest of the peers. In a pure P2P system, peers perform local data management and coordination with their acquaintees, i.e. their one-hop neighbours in the overlay. Thus, data management and coordination is totally decentralized. During the acquaintance procedure between two peers, they exchange information about part of their local schema and create a mediating mapping semi-automatically [1]. The establishment of an acquaintance implies an agreement for the performance of data coordination between the acquaintees based on the respective schema mappings. However, peers do not conform to any kind of data or schema transformation in order to establish acquaintances with other peers, and, hence, participate in the system.

In a large-scale unstructured P2P data management system as described above, joining peers become acquainted to the first randomly available nodes and not to the most useful ones, i.e. the peers that best meet their need for information. Therefore, they have to direct queries not only to their neighbours, but also to a greater part of the system. As a consequence of the lack of global schema, peers express and answer queries on their local schema. Furthermore, the lack of global knowledge deprives peers from the ability to direct their queries to appropriate remote nodes.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 29–37, 2008.

The straightforward procedure for query processing in an unstructured P2P data management system consists of the propagation of the query on paths of bounded depth in the overlay. At each routing step, the query is rewritten to the schema of its new host based on the respective acquaintance mappings. A query may have to be rewritten several times from peer to peer till it reaches nodes that are able to answer it sufficiently in terms of quality but also quantity. It is obvious that the successive rewritings decrease the information held by a query and, thus, also reduce the possibility of accurate query answering. Moreover, it is the case that peers may not be able to sufficiently answer received queries not because their local schema does not match the initial query adequately, but because the incoming rewritten version has been gradually reduced or corrupted. Therefore, the performance of the query processing procedure is degraded during the rewritings on intermediate peers.

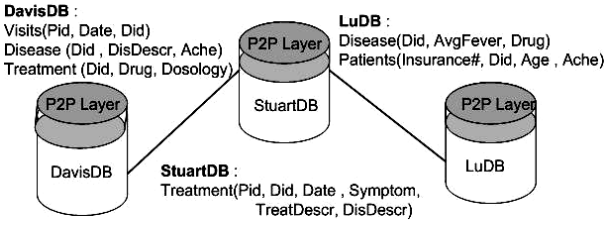


Fig. 1. Part of a P2P system from the health environment

Motivating Example

Envision a P2P system where the participating peers are databases of private doctors of various specialties, diagnostic laboratories and databases of hospitals. Figure 1 depicts a small part of this system, where nodes are: DavisDB - the database of the private doctor Dr. Davis, LuDB - the database of pediatrist Dr Lu and StuartDB – the database of the pharmacist, Mr Stuart. A P2P layer on top of each database is responsible for all data exchange between a peer and its acquaintees. The P2P layer is also responsible for the creation and maintenance of mappings between local schemas during the establishment of acquaintances towards the line of [1]. Moreover, each peer owns a query rewriting and a query-schema matching mechanism.

The local schemas exported by these peers are shown in Figure 1. Suppose that Dr Davis would like to collect from the system general information about patients that have had diseases. He expresses the following query on his database:

```
Qorig:
SELECT V.Pid, D.DisDescr, D.Ache,
T.Drug, T.Dosology
FROM  Disease D, Treatment T, Visits V
WHERE V.Did = D.Did AND D.Did = T.Did
```

Having only one acquaintance, the pharmacist's database, Dr. Davis's database propagates *Qorig* to it. We assume GAV/LAV/GLAV (i.e. Global, Local, Global and Local As View) mappings between acquaintees [2]. We assume the following mapping between DavisDB and StuartDB databases:

```
M_StuartDB_DavisDB:
Treatment(Pid,_,_, Symptom, TreatDescr, DisDescr):-
Visits(Pid,_, Did),Disease(Did, DisDescr, Ache),Treatment(Did,
Drug,_),
where correspondences Symptom = Ache, TreatDescr = Drug are
implied. Thus, the rewritten query on StuartDB is the
following:
QStuartDB_sr:
SELECT T.Pid, T.DisDescr, T.Symptom, T.TreatDescr
FROM Treatment T
```

Obviously, the new query has lost the attributes referring to information about drug dosology, since it cannot be mapped in StuartDB. The node of Mr Stuart passes the rewritten version QStuartDB_sr to Dr Lu with whom he has the following GAV mapping:

```
M_StuartDB_LuDB:
Treatment(Pid,_,_, Symptom,_,_ ):- Disease(Did, AvgFever,_),
Patients(Insurance#, Did,_,_), Age < 13
where correspondences Pid = Insurance#, Symptom = AvgFever are
implied. Thus, the rewritten query on LuDB is the following:
QLuDB_sr:
SELECT P.Insurance#, D.AvgFever
FROM Disease D, Patients P
WHERE D.Did = P.Did, P.Age < 13
```

Clearly, the new query has lost more attributes, which refer to the description of the disease and the respective drug. Moreover, the new query is more restrictive than the original, since it has an additional condition on Age. Finally, it is clear that the Ache attribute of the original query has been poorly rewritten to AvgFever, even though the schema of LuDB contains an attribute that represents the exact same concept. Yet, if Dr Davis was acquainted with Dr Lu, among the supported mappings could be:

```
M_LuDB_DavisDB:
Visits(Pid,_, Did), Disease (Did,_, Ache), Treatment (Did,
Drug,_):-Disease(Did,_, Drug), Patients(Insurance#, Did,_,
Ache),
where the correspondence Pid = Insurance# is implied. Using
the above mapping, Dr Davis would ideally like his query to be
translated as follows:
QLuDB_ideal :
SELECT P.Insurance#, D.Ache, D.Drug
FROM Disease D, Patients P
WHERE D.Did = P.Did
```

The above version overcomes the degradation of successive rewriting in terms of query information loss and further query restriction, as well as the poor matching of the Ache attribute.

We are interested in data exchange issues in pure (i.e. without super-nodes) and unstructured (i.e. without distributed indexes) P2P database systems. Specifically, we are investigating methods for the better fulfilment of peer-database information needs. We are concerned about the provision of more accurate and complete query answers without sacrificing peer autonomy and distorting the system from the P2P paradigm

towards a system with rigid centralized control. Moreover, we aim at solutions that compensate peers according to their activity in the system. Towards this end we are elaborating on exploiting solely the queries posed by each peer in order to discover the interesting sources. Furthermore, we are investigating methods for spreading the knowledge of each peer about schema information to the rest of the system and adapting this experience to the evolution of the peers.

Our Proposal: *GrouPeer*

In this work we develop *GrouPeer*, a system designed to enable accurate query evaluation through semantic overlay clustering and automatic creation and maintenance of semantic groups in relational P2P databases without prior schema or meta-schema information. In *GrouPeer*, nodes individually decide whether to answer the successively rewritten query or automatically rewrite its original version. Requesters evaluate the replies along with the returned rewritings and gradually build mappings with remote, previously not acquainted to, peers. Eventually, peers with similar local schemas become acquainted and clusters are created around active peers. *GrouPeer* intends to make the following contributions:

- Adaptation of classical query rewriting to the needs of P2P databases and successful combination of it with automatic schema matching.
- Investigation of the notion of query similarity in the context of the P2P paradigm and proposition of directions for its quantification
- Provision of a complete methodology for discovering similar peers in an unstructured overlay and gradual clustering of them by utilizing learning through regularly posed query traffic.
- Exploitation of the results of learning so that mappings between remote peers are gradually built based on their specific common interests; this facilitates the acquaintance procedure that is usually performed through human interaction.
- Development of processes that enable the preservation of matching experience that is ordinarily lost due to peer schema alterations.

Coming back to our motivating example, in *GrouPeer*, DavisDB can evaluate Dr Lu's query translations (e.g., suggest that 'Ache'='AvgFever' is not a good correspondence and 'Pid'='Insurance' is a good one) and enable him to gradually improve the quality of its query rewriting. Through mutual iterative evaluations Dr Davis notices the average answer quality from Dr Lu is high enough to add him as an acquaintee. The two nodes create complete mappings between their schemas, a task that GrouPeer greatly facilitates by building on mappings formed during remote query processing.

2 Related Work

We briefly list the most influential works on data exchange problems in the rising P2P database area.

The Chatty Web [3] considers P2P systems that share (semi)-structured information. The authors are concerned about the gradual degradation, in terms of syntax and

semantics, of a query propagated along a network path. This approach considers peers with very simple relational schemas and GAV mappings with their acquaintees. Instead, we are interested in more complex schemas and we consider GAV, LAV or GLAV mappings.

In [4], the authors propose optimization techniques for query reformulation in P2P database systems. They focus on minimizing the rewriting of a query and pruning the propagation path in order to avoid redundant reformulations. While in [4] central knowledge of the system structure is required, our scheme is developed in order to enable operation in a completely decentralized fashion, utilizing the standard lookup operations to refine their local knowledge.

PeerDB [5] features relational data sharing without schema knowledge. Query matching and rewriting is based on keywords. First all nodes within a TTL radius are contacted, returning prospective answer meta-data. The user selects the relevant ones to the query and the selected sources are contacted directly for the results to the various rewritten query versions. Instead, our approach employs an automated technique based on a combination of successive query rewriting and query-schema matching, while it utilizes bandwidth-efficient walks than the costly flooding scheme.

Edutella [6] is a schema-based network that holds RDF data. Peers register the query-types they can answer to mediators, which route queries to appropriate peers. Edutella is an effort towards the solution of the heterogeneity problem of data and services. However, it does not focus on semantic clustering, neither does it propose sophisticated methods for distributing queries to semantically relevant peers.

3 Main Issues

An already explored direction towards the solution of the query degradation problem is the support of 'interest groups', (e.g. [6]), as a communication mean among peers. Interest groups are usually communities predefined in the P2P system that offer one or more representative schemas of the respective information. Joining peers can choose the most suitable for them interest group(s) and create mappings between the schemas they offer and their own. Using this mapping, posed queries are translated on the interest group's schema and travel across the P2P network. Peers that belong to the same interest group and receive the query can use their own mapping to the former, translate the query to the local schema and answer it.

This solution is very clear, simple and sound. However, it does not adhere fully to the P2P paradigm. The latter dictates a system with as autonomous peers as possible that do not adjust in any way to central authorities. Hence, the use of global predefined inalterable schemas as a means to pose queries on various sources leads to dynamic data integration, rather than to P2P systems.

Moreover, we are challenged to develop information fulfillment techniques that apply more to a P2P environment rather than to a distributed database system. The P2P database model is frequently misinterpreted since it combines two conflicting characteristics: on one hand it refers to databases, i.e. sources that comprehend and communicate information in very strict formats and, are traditionally used for utterly disciplined data retrieval; on the other hand the P2P paradigm was born in order to serve broader information needs with looser specifications. Therefore, the P2P

database model dictates systems that comprise sources that store and share only structured data, yet their participation aims to a general benefit from the wide available knowledge in the system.

Our work focuses on solutions that exemplify flexibility and adaptation to the dynamicity of the P2P environment. Specifically, we assume that the main available assets are solely the pair schema mappings between the randomly acquainted peers as well as the queries posed to the system. Moreover, we assume that peers seek for approximate information of various degrees. Our first goal is to develop a technique that enables peers to exploit their queries in order to discover remote peers suitable for establishing acquaintances. Section 5 refers to our initial effort towards this goal.

Furthermore, our intension is to provide the memory-less P2P database system with mechanisms that enable peers to exploit past experience. Since the realization of schema mappings between acquaintees is expensive and the P2P nature deprives peers from static global references, the only way to enhance communication is to reutilize mapping experience. Mappings are dropped due to peer schema evolution and acquaintance transitions. If we could deduce mapping experience before dropping them, then new acquaintances could be guided in order to achieve better mappings with less effort.

4 Evading Successive Query Rewriting

As aforementioned, queries cannot benefit from peers rich in sought information hidden behind poor ones. But what if these peers had the chance to receive and answer the originally posed query? Then, the inquiring node would (a) get better answers to its query, and (b) have the chance to learn about peers with similar interests.

Our work in [7] is a first step towards the exploration of such a solution. It proposes the evasion of successive rewritings on every peer of a query's propagation path, rather than, sometimes hopelessly, trying to ameliorate query reformulation. The presented methodology combines query rewriting and automatic schema matching.

Query Propagation: The key idea is to propagate along the query path not only the successively rewritten query version, but also the original one. In this way, peers that receive this pair of query versions can decide which one to answer.

Query Answering: The peers are assumed to be equipped with a query rewriting mechanism but also an automatic schema-matching tool. The rewriting mechanism is used in order to reform queries expressed on schemas of acquaintees based on the respective mappings. The automatic schema-matching tool is used in order to comprehend and translate queries expressed on unknown schemas. A peer uses the following metric in order to make this decision:

$$\mathrm{Cov}_p(Q_{v2}, Q_{v1}) = \mathrm{Msim}_p(Q_{v1}, Q_{v2}) \cdot \theta_p(Q_{v1})$$

The above metric estimates the *coverage* of the target (rewritten) query Q_{v2} on the source query Q_{v1}. The metric takes into account the similarity of the two queries weighted by the confidence of the peer about the accuracy of the overall estimation procedure. Specifically, the function $\mathrm{Msim}_p(Q_{v1}, Q_{v2})$ measures the similarity

between the source, Q_{v1}, and the rewritten query, Q_{v2}. In other words, $Msim_p$ is used by peer P to estimate how much of Q_{v1} can be answered by Q_{v2}. Thus, query similarity is subjective and can refer to the query structure and/or to the query results. However, the implementation of Msim depends on the individual understanding of the above measure definition of each peer and, consequently, it depends on the schema matching method used.

Beyond Msim, each peer-database has a confidence measure θ that characterizes the overall ability of the peer to be able to `guess' the correct rewritten version of a query for which it has no schema information. A confidence parameter depends on internal features of a peer and on the query itself:

- the peer's estimation about its schema matching ability.
- the structure of the query: specifically, the amount of information given by the structure of the query for the schema on which it is expressed, so that there can be a certain degree of guarantee that a respective rewriting is accurate.

Therefore, θ expresses the guarantee for correctness and completeness of a query rewriting produced by an automatic schema-matching tool. Thus, θ is a function of the evaluated query Q and the structure of the function depends on the peer P: $\theta = \theta_p(Q)$. The peers are advised to use low θ values for queries for which they have no schema information and they receive for the first time (or even more for queries initiated by peers of which they have no previous experience). In this way, peers are encouraged to select to answer the successively rewritten queries unless it is so much degraded that it is worth trying automatic matching for the original query.

Answer Evaluation: The peer sends back to the query initiator a packet with the answer-related information. Then, the initiator can make its own evaluation of the answer and, furthermore about the relative ness of its interests to the content of the answering peer. The initiator evaluates the result according to:

- its estimation of how easily other peers can correctly andcompletely rewrite Q,
- and its requirement for accurate answers to Q.

Building Mappings: The query initiator announces its evaluation of the result to the answering peer. Good correspondences can be used to build mappings between the two peers and bad ones can be avoided in future query rewriting. In this way if the latter receives in the near future the same (or even a similar) query from the same initiator, it can adjust the respective automatic rewriting and θ value to the initiator's indications. Furthermore, the adjustment step of a θ value influences the number of recursions of the feedback procedure. Hence, a low step gives the opportunity to the peer to get many, potentially good, evaluations from the requester and ameliorate its matching procedure gradually and to a finer degree. Accordingly, *GrouPeer* achieves:

- the gradual training in automatic schema matching for queries initiated by specific peers
- the discovery of nodes concealed in the network that are convenient for acquaintees using as a mean only the queries posed in the P2P system.

Therefore, this methodology can be used in order to progressively *cluster* the overlay so that peers with common interests are close or acquainted.

Each peer can periodically evaluate peers that have answered queries on its behalf. Accordingly, the query initiator can decide if the answering peer has given overall good answers or not, or if this decision is still pending.

Simulation experimental results such as those in Figure 2 (for more details see [7]) show that the proposed method achieves to surpass the deficiencies of successive query rewriting and discover remote sources convenient for information supply that cannot be spotted otherwise.

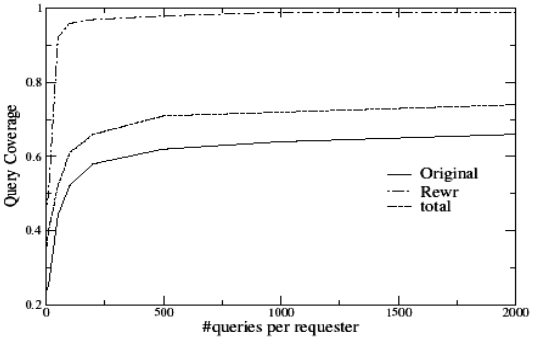


Fig. 2. Query coverage for answers to the rewritten and original queries as the number of queries increases

5 Conclusions and Future Work

Our work focuses on exchange issues in P2P data management systems. P2P database systems suffer often by peer isolation: peers reside in neighbourhoods of dissimilar interests and schemas. The scope of our research is to provide peers with means that will enable them to retrieve interesting information although they may be initially badly situated in the system.

In Section 4 we described our first attempt towards the solution of the query degradation problem in P2P databases in the absence of global schema information. By allowing peers to select to answer the appropriate rewritten version of a query, they discover remote peers on query propagation paths that are rich in interesting information but veiled by poor path predecessors. Using these discoveries, nodes sharing similar information are gradually interconnected, increasing the quality of the returned results. Our solution is specifically suited for dynamic, unstructured environments, since it is adaptive, bandwidth-efficient and operates in a complete decentralized manner. Currently we are investigating query similarity measures to be used as guidelines for peers in order for each one to create its own such measure. Also, we are investigating the structure of the Θ parameter.

Furthermore, the described procedure allows for the discovery of schema mappings between two peers. We intend to explore means for the reuse of the revealed schema information of a pair of peers by other existing or new members of the system.

Moreover, we intend to investigate extensions of the proposed process so that it adapts to schema evolution of peers. Particularly, our goal is to ensure that the estimations about the schema of distant peers do not have to be dropped altogether in case of schema alterations.

Beyond this, we are interested in enabling the preservation of matching experience that is ordinarily lost due to peer schema alterations. In this way we can achieve incremental comprehension of the evolving P2P system.

References

1. Kantere, V., Kiringa, I., Mylopoulos, J., Kementsientidis, A., Arenas, M.: Coordinating P2P Databases Using ECA Rules. In: Intern. Workshop of DBISP2P (2003)
2. Lenzerini, M.: Data Integration: A Theoretical Perspective. Tutorial, ACM PODS (2002)
3. Aberer, K., Cudre-Mauroux, P., Hauswirth, M.: The Chatty Web: Emergent Semantics Through Gossiping. In: WWW Conference (2003)
4. Tatarinov, Halevy, A.: Efficient Query Reformulation in Peer-Data Management Systems. In: ACM SIGMOD (2004)
5. Ooi, B., Shu, Y., Tan, K.L., Zhou, A.Y.: PeerDB: A P2P-based System for Distributed Data Sharing. In: ICDE (2003)
6. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T.: Edutella: A p2p networking infrastructure based on rdf. In: WWW (2002)
7. Kantere, V., Tsoumakos, D., Roussopoulos, N.: Querying Structured Data in an Unstructured P2P System. In: ACM Intern. Workshop on WIDM (2004)

On Enhancing Query Optimization in the Oracle Database System by Utilizing Attribute Cardinality Maps

B. John Oommen and Jing Chen

School of Computer Science, Carleton University, Ottawa, Canada, K1S 5B6
{oommen,jchen8}@scs.carleton.ca

Abstract. Central to the process of query optimization in all real-life modern-day Database Management Systems (DBMS) is the use of histograms. These have been used for decades in approximating query result sizes in the query optimizer, and methods such as the Equi-Width and Equi-Depth histograms have been incorporated in all real-life systems. This is because histograms are simple structures, and can be easily utilized in determining efficient Query Evaluation Plans (QEPs). This paper demonstrates how we can incorporate two recently-developed histogram methods into the ORACLE real-life DBMS. These two new histograms methods were introduced by Oommen and Thiyagarajah [1], and called the the Rectangular Attribute Cardinality Map (R-ACM), and the Trapezoidal Attribute Cardinality Map (T-ACM).

The superiority of the R-ACM and the T-ACM in yielding more accurate query result size estimates has been well demonstrated, and the resulting superior QEPs for a theoretically-modeled database was shown in [2]. In this paper we make a "conceptual leap" and demonstrate how the ACMs can be incorporated into a *real-life* DBMS. This has been done by designing and implementing a prototype which sits on top of an ORACLE 9*i* system. The integration is achieved in C/C++ and PL/SQL, and serves as a prototype "plug-in" to the ORACLE system, since it is fully integrated and completely transparent to users. The superiority of utilizing the ACM histograms is rigorously validated by conducting an extensive set of experiments on the TPC-H benchmark data sets, and by testing on equi-select and equi-join queries. The entire set of experimental results obtained by integrating the underlying algorithms into the ORACLE query optimizer can be found in [3].

Keywords: Query Optimization, Histogram-based Algorithms, Attribute Cardinality Maps, ORACLE DBMS, TPC-H benchmark.

1 Introduction

In this paper, we[1] we consider the well-known problem of query optimization[2], and propose how we can incorporate some recently-proposed histogram-based methods to enhance a fully-developed real-life database system.

[1] The first author also holds an *Adjunct Professorship* with the Department of Information and Communication Technology, Agder University College, Grooseveien 36, N-4876 Grimstad, Norway.

[2] A *brief, preliminary* version of this paper was presented at ICEIS-2006, the 2006 International Conference on Enterprise Information Systems, in Cyprus, in May 2006. *This talk was a Plenary/Keynote Talk of the Conference.*

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 38–71, 2008.

Query optimization can usually be achieved in two different ways: on the database server side, and on the application side. The former method depends on the architecture and design of the core of the DBMS, while the latter depends largely on the applications themselves. The design of the database query optimizer, and the algorithms it utilizes, plays an important role in the efficiency of the DBMS. As in the case of various algorithms used in the DBMS, histograms have been studied extensively and utilized in the design and implementation of database systems. They are widely used in selectivity estimation for database query optimization, and in approximating query result sizes. A great deal of research work has been done in this area, and we shall not attempt to survey the field here. Various histogram-based algorithms were proposed in [4], [5], [6], [7], and [8]. Also, the optimization of these algorithms was proposed in [9], [7], and [8]. Additionally, the use of sampling in this problem has been addressed in [10] and [11]. New, error-bounded algorithms were proposed in [12], [2], [13], [14], [15], and [1], which form the core of the strategies presented in this paper. While the salient details of these are highlighted presently, a *brief* survey of the pertinent results will be presented in a later section. A more detailed survey is found in [3] and [1].

The accuracy of the estimates of query result sizes is of fundamental importance in query optimization. Also, it is well known that it has direct influence on choosing optimal QEPs, including the selection of the access paths, the join orders, and the join methods. For example, it has been shown in [16] that errors of the estimates of query result sizes may grow *exponentially* with the number of joins. In other words, it is (both theoretically, and in real-life systems) one of the most important aspects useful for determining the effectiveness of selecting a hopefully optimal QEP.

In this paper, we address the problems of estimating query result sizes in a *real-life* database system, namely the ORACLE DBMS. As mentioned earlier, Oommen and Thiyagarajah had introduced two new histogram-like techniques, namely the Rectangular Attribute Cardinality Map (R-ACM), and the Trapezoidal Attribute Cardinality Map (T-ACM). They, and later Oommen and Rueda [2], had also clearly demonstrated the superiority of these in yielding more accurate query result size estimates for theoretical and real-life databases. Oommen and Rueda [2] later also showed that for a theoretically-modeled database, the *actual* QEPs obtained by invoking ACMs in estimating the query result sizes were superior to those obtained by the traditional methods. In spite of the ACM-related work that was done earlier, the question of how the ACMs would perform in a real-life DBMS was unanswered. This is the primary focus of this paper. In this paper we demonstrate how they can be incorporated into a *real-life* DBMS. This has been done by designing and implementing a prototype which sits on top of an ORACLE 9*i* system. The paper also reports some results obtained by extensive and rigorous testing using the TPC-H benchmark database.

This paper is organized as follows: In Section 2 we *briefly* review some pertinent previous work done on using various histogram-based algorithms, which are either widely used in commercial DBMSs, or are relatively new, and have been proven to possess excellent characteristics. Section 3 briefly introduces the architecture of the implementation of the ORACLE "plug-in", and the methods used to incorporate the new histogram

algorithms. In Section 4 we present the test bed used in the entire experiments, and the experimental results obtained by utilizing the R-ACM and the T-ACM in the ORACLE DBMS. We then demonstrate their superiority by comparing the estimates and QEPs with those obtained by using the Equi-depth algorithm. In order to not distract from the thrust of the paper, screen shots from the QEPs that compare the currently-used histogram algorithms and the new algorithms for most of the relevant sample queries, are presented in Figure 3. It is pertinent to mention that the ACMs seem to yield superior QEPs for all the types of queries analyzed. Section 5 concludes the paper and indicates problems of interest for further study. Of course, the business potential is phenomenal.

1.1 Contributions of This Paper

In this paper, we have investigated the issue of query optimization from the point of view of the internal aspects of a *real-life* DBMS, and not merely from the perspective of the accuracy of the query result sizes. We have studied various histogram-based algorithms used for approximating query result sizes, and in particular, how the optimal QEP is estimated in the ORACLE DBMS. Of course, as mentioned earlier, the intention was to investigate the performance of the ACMs in a real-life DBMS. To render the comparison meaningful, as part of the novel work in this paper, we have extended the capabilities of the ORACLE DBMS by incorporating the two new ACM-based algorithms into the query optimizer. The integration is implemented in C/C++ and PL/SQL, and serves as the prototype of a "plug-in" to the ORACLE system, since it is fully integrated and is completely transparent to users. Thus, the user specifies a query, and has the option of determining the QEP using either the standard ORACLE histogram method, or invoking the ACM as the histogram, and *thus* reporting the QEP. The superior performance of these two algorithms is validated by experiments conducted on the TPC-H benchmark database with simplified equi-select and equi-join queries. Additionally, their superiority is also verified by comparing the *actual* QEP results obtained by using them and by using the Equi-depth, which is the traditional method most widely used in ORACLE, and in most current commercial DBMSs.

2 Basic Structure of a Script

2.1 Histogram Methods

Histogram-based techniques are the most widely used methods among the non-parametric estimates used in query optimization. For a given distribution of the attribute, these techniques approximate the estimates by a pre-computed histogram. Generally, the histogram is represented by the attribute values which are divided into *buckets*, each of them containing an arbitrary number of values, reflected by the frequencies of the attribute values. To minimize the storage requirements for the algorithms, typically, only the interval bounds and the frequency counters are stored. However, the strategy by which it is applied is, clearly, algorithm dependent. We briefly review some of the

relevant histogram-based methods used in query optimization. *The review is, by necessity, brief and focused on those relevant to this current paper.*

The Equi-width Histogram. The use of the Equi-width histogram in databases was predominantly advocated by Piatetsky-Shapiro and Connell [17]. As its name implies, the Equi-width histogram has, as approximately as possible, the same width for each bucket, while the frequencies or the heights are variable. The frequency in each bucket is calculated by taking the *uniform frequency assumption*.

The drawback of the Equi-width method is obvious since it utilizes the *uniform frequency assumption*. It will not yield a good performance if the attribute values are skewed, sparse, and if the bucket boundaries are not determined appropriately. Moreover, the statistics literature offers little help in solving the problem without some knowledge of the distribution [18]. The latter paper also shows the disadvantages of the Equi-width method when compared to other reported statistical methods.

The Equi-depth Histogram. The Equi-depth histogram is the most widely-used one in the commercial database systems. Basically, this technique consists of dividing the given attribute values into buckets such that each of them has, *as approximately as possible*, the same number of frequencies. The frequency of a required attribute value is approximated by dividing the population in the bucket by the number of attribute values placed in that bucket. The properties of the Equi-depth method are described in detail in [19] and [20]. Although this histogram has vast applications, it is really *not* efficient for the case when the data is sparse, because, as most of the frequency values will be zero, the bucket sizes can be extremely large. It has also been shown that the errors caused by Equi-depth estimates can be very large, and that these errors can increase in an unbounded manner as the depth of the query tree increases. This is also our experience.

In the interest of brevity, we will not survey or discuss any of the other existing histogram-based methods. Rather, we will briefly mention the ACM technologies, which are the principal focus of this paper.

The R-ACM. In 1999, Oommen and Thiyagarajah [1] proposed two new histogram-based methods based on the concept of Attribute Cardinality Maps. In particular, they introduced the Rectangular ACM (the R-ACM) and the Trapezoidal ACM, (the T-ACM), which improved the effectiveness of histograms in *estimating* query result sizes. In Thiyagarajah's Ph.D. Thesis [1], the mathematical foundation of both the R-ACM and the T-ACM were provided, and formally proven. The R-ACM is a modified form of the histogram. It achieves an excellent error control by disallowing large frequency differences within a single bucket. The essence of the R-ACM is that the error is bounded by minimizing the variation from the running mean within a bucket. The R-ACM has a variable bucket width, and a variable number of tuples associated with each bucket. Therefore, the bucket boundaries and the height for each bucket are parameters to be determined. Based on a rule that aims at minimizing the estimation error, the sector widths can be generated over given frequency distributions of attribute values. Another highlight of the R-ACM histogram is that it is catalogue based, which

Table 1. Notations Used in the Paper

Symbol	Explanation
x_i	Number of tuples in attribute X for the i^{th} value of X.
$E(X_i)$	Expected number of tuples in attribute X for the i^{th} value of X.
n_j	No of tuples in the j^{th} sector of an ACM.
l_j	No of distinct values in the j^{th} sector.
s	Number of sectors in the ACM.
τ	Allowable tolerance for an R-ACM
ξ	Size of a relation.
N	Number of tuples in the relation.

means that it can be pre-computed, and will thus not incur much I/O overhead during run time.

The R-ACM can be either one-dimensional or multi-dimensional, depending on the number of attributes being mapped. We shall start by presenting a few concepts related to the one-dimensional case.

We present in Table 1 the notations that will be used to describe the R-ACM.

Definition 1. A One dimensional Rectangular ACM: *Let $\mathcal{V} = \{v_i : 1 \leq i \leq |\mathcal{V}|\}$, where $v_i < v_j$ when $i < j$, be the set of values*[3] *of an attribute X in relation R. Let the value set $\mathcal{V}$ be subdivided into s number of sector widths according to the range partitioning rule described below. Then the Rectangular Attribute Cardinality Map of attribute X is an integer array in which the j^{th} index maps the number of tuples in the j^{th} value range of the set $\mathcal{V}$ for all j, $1 < j \leq s$.*

Range Partitioning Rule: Given a desired tolerance value τ for the R-ACM, the sector widths, $l_j, 1 \leq j \leq s$, of the R-ACM should be chosen such that for any attribute value X_i, its frequency x_i does not differ from the running mean of the frequency of the sector by more than the tolerance value τ, where the running mean is the mean of the frequency values examined so far in the current sector.

For example, consider the frequency set $\{8,6,9,7,19,21,40\}$ corresponding to the attribute values $\{X_0,X_1,X_2,X_3,X_4,X_5,X_6\}$ of an attribute X. Using a tolerance value $\tau = 2$, the attribute value range will be partitioned into the three sectors, $\{8,6,9,7\}$, $\{19,21\}$, $\{40\}$ with sector widths of 4, 2, and 1 respectively.

Using the *range partitioning rule*, Algorithm `Generate_R-ACM` partitions the value range of the attribute X into s variable width sectors of the R-ACM.

[3] In this work, only ordinal numerical values are considered for attributes. It is possible to convert non-numeric attributes (symbolic, scalar-typed, fuzzy etc.) into ordinal numbers using a mapping function. This includes most type of attribute values. For non-ordinal data (for example, "real valued" data), we have proposed another type of structure called the Trapezoidal Attribute Cardinality Map (T-ACM) explained later.

Algorithm 1. Generate_R-ACM

```
Input: tolerance τ, frequency distrib. of X as A[0...L−1]
Output: R-ACM
begin
  Initialize_ACM;      /* set all entries in ACM to zero */
  current_mean := A[1]; j := 0;
  ACM[j] := A[1];
  for i:=1 to L−1 do      /* for every attribute value */
    if abs(A[i]− current_mean) < τ
      ACM[j] := ACM[j] + A[i];
      current_mean :=
      (current_mean*i + A[i])/(i+1); /* running mean */
    else begin
      l_j := i − 1;      /*set the sector width */
      j++;      /* move to next sector */
      current_mean := A[i];
      ACM[j] := A[i];
    end;
end;
end Algorithm
```

The input to the algorithm are the tolerance value τ for the ACM and the actual frequency distribution of the attribute X. The frequency distribution is assumed to be available in an integer array A, which has a total of L entries, one for each of the L distinct values of X. For simplicity reasons, we assume that the attribute values are ordered integers from 0 to $L-1$. The output of the algorithm is the R-ACM for the given attribute value set.

It is obvious that Algorithm, `Generate_R-ACM` generates the R-ACM corresponding to the given frequency value set.

Assuming that the frequency distribution of X is already available in array A, the running time of the Algorithm `Generate_R-ACM` is $O(L)$ where L is the number of distinct attribute values.

Since the ACM only stores the count of the tuples and not the actual data, it does not incur the usually high I/O cost of having to access the base relations from secondary storages. Secondly, unlike the traditional histogram-based or other parametric and probabilistic counting estimation methods currently in use, the ACM does not use sampling techniques to approximate the data distribution. Each cell of the ACM maintains the *actual* number of tuples that fall between the boundary values of that cell, and thus, although this leads to an approximation of the density function, there is no approximation of the number of tuples in the data distribution.

The one-dimensional R-ACM as defined above can be extended to a multi-dimensional one easily to map an entire multi-attribute relation. A multi-dimensional ACM can also be used to store the multi-dimensional attributes that commonly occur in geographical, image, and design databases.

The following are the pertinent results applicable to R-ACM.

Lemma 1. *The probability mass distribution for the frequencies of the attribute values in an R-ACM can be approximated to be a Binomial distribution with parameters* $(n, \frac{1}{l})$. □

The following results must all be read in the context of the previous Lemma. In other words, we shall assume the distribution of the attribute values to follow the one specified by Lemma 1.

Theorem 1. *For a one-dimensional rectangular ACM, the maximum likelihood estimate of the number of tuples for a given value* X_i *of attribute X is given by,*

$$\hat{x}_{ML} = \frac{n}{l}$$

where n is the number of tuples in the sector containing the value X_i *and l is the width of that sector.* □

Theorem 2. *For a one-dimensional rectangular ACM, the maximum likelihood estimate of the number of tuples for a given value* X_i *of attribute X falls within the range of,*

$$\frac{(n+1)}{l} - 1 \leq \hat{x}_{ML} \leq \frac{(n+1)}{l},$$

where n is the number of tuples in the sector containing the value X_i *and l is the width of that sector.* □

Lemma 2. *For a one-dimensional rectangular ACM, the expected number of tuples for a given value* X_i *of attribute X is* $E(X_i) = n/l$*, where n is the number of tuples in the sector containing the value* X_i*, and l is the width of that sector.* □

Lemma 3. *The variance of the frequency of an attribute value X in sector j of an R-ACM is,*

$$Var(X) = \frac{n_j(l_j - 1)}{l_j^2} \tag{1}$$

□

Lemma 4. *The sector variance of the* j^{th} *rectangular ACM sector is,*

$$Var_j = \frac{n_j(l_j - 1)}{l_j} \tag{2}$$

□

Lemma 5. *The variance of an R-ACM is given by,*

$$Var(ACM) = \sum_{i=1}^{s} Var_i \tag{3}$$

where s is the number of sectors in the ACM. □

Lemma 6. *The error,* ε*, resulting from a self-join of relation R on attribute X using a rectangular ACM is given by,*

$$\varepsilon = Var(ACM) + \sum_{j=1}^{s} \left\{ \sum_{k=1}^{l_j} x_k^2 - \frac{n_j^2 + n_j l_j - n_j}{l_j} \right\}. \qquad \square$$

Theorem 3. *The variance of a rectangular ACM corresponding to attribute X is,*

$$Var(ACM) = N - \sum_{j=1}^{s} \frac{n_j}{l_j}. \tag{4}$$

$\square$

Lemma 7. *A decreasing R-ACM sector is "least uniform", if and only if*

$$x_k = a - \sum_{i=1}^{k-1} \frac{\tau}{i} \qquad \text{for } 1 \le k \le l_j. \qquad \square$$

Lemma 8. *An increasing R-ACM sector is "least uniform", if and only if*

$$x_k = a + \sum_{i=1}^{k-1} \frac{\tau}{i} \qquad \text{for } 1 \le k \le l_j. \qquad \square$$

Theorem 4. *If the value* X_i *falls in the* j^{th} *sector of an R-ACM, then the number of occurrences of* X_i *is,*

$$\frac{n_j}{l_j} - \left| \tau \left[\ln \left(\frac{l}{i-1} \right) - 1 \right] \right| \le x_i \le$$

$$\le \frac{n_j}{l_j} + \left| \tau \left[\ln \left(\frac{l}{i-1} \right) - 1 \right] \right|$$

where n_j *and* l_j *are the number of tuples and the sector width of the* j^{th} *sector.* $\square$

More properties of the R-ACM are given in [14] and [1], and its power in minimizing error estimates for query result sizes, and in determining nearly-optimal QEPs for theoretical databases is found in [2] and [1].

The T-ACM. The Trapezoidal Attribute Cardinality Map (T-ACM) is the other new histogram-based method proposed by Oommen and Thiyagarajah in [15] and [1]. We briefly describe it below.

The Trapezoidal Attribute Cardinality Map (T-ACM) is also a non-parametric histogram-based estimation technique. The T-ACM generalizes the R-ACM from a "step" representation to a "linear" representation. It has equal bucket widths and variable

linear-sloped bucket heights. It is a modified form of the Equi-width histogram where each histogram partition is a trapezoid instead of a rectangle. In fact, the T-ACM is obtained by replacing each of the rectangular sectors of the Equi-width histogram by a trapezoid. The beginning and the ending frequency values of each bucket is chosen so that the area of the resulting trapezoid will be equal to the area of the "rectangle" of the histogram it is replacing.

The T-ACM can be either a one-dimensional or a multi-dimensional depending on the number of attributes being mapped. Again, we begin by presenting some concepts related to the one-dimensional case.

A trapezoidal ACM is a modified form of the equi-width histogram where each histogram partition is a trapezoid instead of a rectangle. In fact, the trapezoidal ACM is obtained by replacing each of the rectangular sectors of the equi-width histogram by a trapezoid. The beginning and ending frequency values of each trapezoid sector is chosen so that the area of the resulting trapezoid will be equal to the area of the "rectangle" of the histogram it is replacing.

Definition 2. A One dimensional Trapezoidal ACM: *Let $\mathcal{V} = \{v_i : 1 \leq i \leq |\mathcal{V}|\}$, where $v_i < v_j$ when $i < j$, be the set of values of an attribute X in relation R. Let the value set $\mathcal{V}$ be subdivided into s equi-width sectors, each having sector width, l. We approximate each equi-width sector by a trapezoid in which the j^{th} trapezoid is obtained by connecting the starting value, a_j, to the terminal value, b_j, where the quantities a_j and b_j satisfy:*

(a) The starting value a_1 is a user-defined parameter.
(b) For all $j > 1$, the starting value of the j^{th} trapezoid, a_j, is the terminal value of the $(j-1)^{st}$ trapezoid, b_{j-1}.
(c) The area of the j^{th} trapezoid exactly equals the area of the j^{th} equi-width sector from which the exact computation of the quantity, b_j, is possible.

Then the Trapezoidal Attribute Cardinality Map of attribute X with initial attribute value X_1 and width l is the set $\{(a_i, b_i) | 1 \leq i \leq s\}$.

The motivation for proposing the trapezoidal ACM for density approximation (and the query result size estimation) originates from considering the various techniques used in numerical integration. Finding the result size of a selection query on a range-predicate can be considered as a discrete case of finding the area under a curve. Thus any numerical integration technique used to find the area under a curve will fit our purpose well. Though more accurate and sophisticated methods such as Simpson's Rule exist, since the trapezoidal method is relatively easy to use in a DBMS setting and is much superior to the traditional equi-width and equi-depth histograms currently in use, we have opted to use the trapezoidal method. In addition to providing more accurate result estimation on selection queries on range predicates, it also gives better results on equality-match predicates.

Unlike the R-ACM, where the sector widths are variable, the sector widths of a T-ACM are all equal. Each sector or cell of a T-ACM stores the frequency values of the first and last attribute values in that sector, in addition to the number of tuples in

the sector. Algorithm `Generate_T-ACM` partitions the value range of the attribute X into s equal width sectors of the T-ACM. The input to the algorithm is the number of partitions, s. The frequency distribution is assumed to be available in an integer array A, which has a total of L entries for each of the L distinct values of X. For simplicity, we assume that the attribute values are ordered integers from 0 to $L-1$. The output of the algorithm is the T-ACM for the given attribute value set. Since choosing the starting frequency value of the first trapezoidal sector is important for obtaining the subsequent a_j's and b_j's, we briefly discuss it below.

As we shall see later, from Lemmas 9 and 10, if the frequency of the first attribute value of the first sector of a T-ACM is known, the subsequent a_j's and b_j's can be easily obtained. The problem of obtaining an optimal starting frequency for building a T-ACM is still open and is currently being investigated. Below we have listed some of the methods that can be used to obtain this quantity:

(1) a_1 is a user-defined frequency value.
(2) a_1 is obtained using the average of all the frequencies in the given attribute value domain.
(3) Use the frequency value from (2) above as the starting frequency of the first sector and compute all the a_j's and b_j's in a *left-to-right* manner. Again use the frequency value from (2) above as the terminal frequency of the last sector and compute all the a_j's and b_j's in a *right-to-left* manner. One possibility is to assign a_1 to be the average of the first frequency values resulting from these two methods.

Before presenting the `Generate_T-ACM` algorithm that generates a T-ACM, we shall first present two lemmas that are used in this algorithm.

Lemma 9. *For each sector in the T-ACM, the number of tuples, n_j, is equal to,*

$$n_j = \left(\frac{a+b}{2}\right).l,$$

where a,b are the frequencies of the first and last attribute value in the sector, and l is the number of distinct values in the sector. □

This lemma is important because ensuring that n_j is close to $(a+b)l/2$ would provide us the desired accuracy using trapezoidal approximation.

Let a_j be the frequency of the first attribute value in the j^{th} sector. The first frequency value of the first sector, a_1 can be chosen to be either the actual frequency of the attribute value (i.e: $a_1 = x_1$) or the average frequency of the entire attribute value range (i.e: $a_1 = \frac{N}{sl}$). The subsequent values for $a_j, 2 \leq j \leq s$, do not need to be stored explicitly and can be obtained from Lemma 10.

Lemma 10. *If the frequency of the first attribute value of the first sector of a T-ACM is a_1, then the frequency of the first attribute value of the j^{th} T-ACM sector, $a_j, 2 \leq j \leq s$, is given by,*

$$a_j = (-1)^{j-1}\frac{2}{l}\left\{a_1 + \sum_{k=1}^{j-1}(-1)^k n_k\right\}$$

where n_k is the number of tuples in the k^{th} sector. □

Algorithm 2. Generate_T-ACM

Input: No of sectors, s,
frequency distrib.of X as
$A[0 \ldots L-1]$
Output: T-ACM
begin
Initialize_ACM; /* set all entries in ACM to zero */
$ACM[1].a := \frac{\sum_{i=0}^{L-1} A[i]}{L}$; /* set a_1 to average frequency */
for j := 1 **to** s **do** /* for every sector */
for i := 1 **to** l **do** /* for every attrib.value */
$ACM[j].n := ACM[j].n +$
$+A[(j-1)*l+i]$;
end; { for }
if $(j > 1)$ **then**
$ACM[j].a := ACM[j-1].b$;
$ACM[j].b := 2*ACM[j].n/l - ACM[j].a$;
end; { for }
end;
EndAlgorithm Generate_T-ACM

In practice, we can obtain a_j easily from a_{j-1} as shown in the Algorithm 2. We also obtain a_1 by averaging the frequency values of the entire attribute range. Note that each entry of the ACM array is a record with three fields, namely n, a, b, which store the number of tuples, the frequency of the first value and the frequency of the last value in the sector respectively.

It is obvious that Algorithm Generate_T-ACM generates the T-ACM corresponding to the given frequency value set. Assuming the frequency distribution of X is already available in array A, the running time of the algorithm Generate_T-ACM is $O(L)$ where L is the number of distinct attribute values.

Algorithm 3. Implement_T-ACM

Input: (i) Equi-width histogram H_A with sector width l.
(ii) Starting frequencies of every 2^{nd} sector.
Output: T-ACM with sector width $l/2$.
begin
Merge every two adjacent sectors of H_A to get H_B;
Generate T-ACM, T_A from H_B.
Estimate frequencies of T_A's middle attribute values using Lemma 9.
Generate T_B from T_A using frequencies obtained from the last step.
Estimate frequencies of T_B's middle attribute values Lemma 9.
Generate T_C from T_B using frequencies obtained from the last step.
end;
EndAlgorithm Implement_T-ACM

Since the T-ACM is based on the trapezoidal rule of numerical integration, we expect it to be more accurate than the corresponding equi-width histogram of the same sector width. We shall now describe a process by which we can obtain a T-ACM that is even much more accurate than the T-ACM generated by the above algorithm `Generate_T-ACM`.

Let us assume that T_A is the T-ACM derived from the equi-width histogram, H_A. Also assume that the frequencies of the starting attribute value of *every second* histogram sector are available. Observe that this can be computed in $O(s)$ time (as opposed to $O(L)$), where s is the number of sectors. Then we can generate a T-ACM which has a sector width that is *half* of the sector width of H_A and is much more superior than the initial T-ACM, T_A. The strategy to achieve this is given in the algorithm, `Implement_T-ACM`, below.

The following are the pertinent results applicable to T-ACM.

Lemma 11. *The probability of a given value X_i occurring in the trapezoidal sector can be approximated to be:*

$$p_{X_i}(x_i) = \frac{a_j}{n_j} + \frac{2(n_j - a_j l)}{n_j l(l-1)}.i \qquad 1 \le i \le l-1 \tag{5}$$

where a_j is the frequency for the first attribute value in the j^{th} sector. □

Lemma 12. *The probability mass distribution for the frequencies of the attribute values in a T-ACM can be approximated to be a Binomial distribution with parameters $(n, p_{X_i}(x_i))$.* □

The following results must all be read in the context of the previous Lemma. In other words, we shall assume the distribution of the attribute values to follow the one specified by Lemmas 11 and 12.

Theorem 5. *For a one-dimensional trapezoidal ACM, the maximum likelihood estimate of the number of tuples for a given value X_α of attribute X in the k^{th} T-ACM sector is given by,*

$$\hat{x}_{ML} = a_k + \frac{2(n_k - a_k l)}{l(l-1)}.z_\alpha$$

where n_k is the number of tuples in the k^{th} T-ACM sector, a_k is the frequency of the first attribute value in the k^{th} sector, l is the number of distinct attribute values (or width) of the T-ACM sectors and X_α is the z_α^{th} value in the T-ACM sector. □

Theorem 6. *For a one-dimensional trapezoidal ACM, the maximum likelihood estimate of the number of tuples for a given value X_i of attribute X falls within the range of,*

$$\frac{a_k}{n_k} + \frac{2(n_k - a_k l)}{n_k l(l-1)}.z_\alpha(n_k+1) - 1 \le \hat{x}_{ML} \le$$
$$\le \frac{a_k}{n_k} + \frac{2(n_k - a_k l)}{n_k l(l-1)}.z_\alpha(n_k+1),$$

where a_k is the frequency of the first attribute value in the k^{th} sector, n_k is the number of tuples in the k^{th} sector containing the value X_i and l is the width of that sector. □

Lemma 13. *Using a trapezoidal approximation, the expected number of tuples for a given value X_i of attribute X is,*

$$E(X_i) = a_j + \frac{2(n_j - a_j l)}{l(l-1)}.i,$$

where n_j is the number of tuples in the sector which contains value X_i and l is the sector width. The quantity a_j is the number of occurrences of the first attribute value in the j^{th} sector. □

Lemma 14. *The variance of the frequency of an attribute value X_i in sector j of a trapezoidal ACM is,*

$$Var(X_i) = n_j p_i (1 - p_i), \text{ where}$$
$$p_i = \frac{a_j}{n_j} + \frac{2(n_j - a_j l)}{n_j l(l-1)}.$$ □

Lemma 15. *The sector variance of the j^{th} trapezoidal ACM sector is,*

$$Var_j = n_j - \frac{a_j(l+1)(a_j l - 2n_j)}{3n_j(l-1)} - \frac{2n_j(2l-1)}{3l(l-1)}$$

where a_j is the frequency of the first attribute value in the sector, n_j is the number of tuples in the sector and l is the sector width. □

Lemma 16. *The variance of a T-ACM is given by,*

$$Var(ACM) = \sum_{j=1}^{s} Var_j$$

where s is the number of sectors in the T-ACM, and Var_j is the sector variance given in Lemma 15. □

Lemma 17. *The error, ε, resulting from a self-join of relation R on attribute X using a trapezoidal ACM is given by,*

$$\varepsilon = \sum_{j=1}^{s} \left(\sum_{k=1}^{l} x_k^2 - n_j^2 + n_j Var_j \right)$$

where s is the number of sectors in the T-ACM, and n_j is the number of tuples in the j^{th} sector and Var_j is the variance of the j^{th} sector given in Lemma 15. □

Theorem 7. *The worst-case error, ε, in estimating the equality select operation, $\sigma_{X=X_\alpha}(R)$ in a T-ACM using the maximum likelihood estimate is given by,*

$$\varepsilon = \begin{cases} a_k + \frac{2(n_k - a_k l)}{l(l-1)} z_\alpha & \text{if } z_\alpha < \frac{l(l-1)(n_k - 2a_k)}{4(n_k - a_k l)}, \\ n_k - a_k - \frac{2(n_k - a_k l)}{l(l-1)} z_\alpha & \text{if } z_\alpha \geq \frac{l(l-1)(n_k - 2a_k)}{4(n_k - a_k l)}. \end{cases}$$

where the attribute value X_α is in the z_α^{th} position within the k^{th} T-ACM sector. □

Theorem 8. *The worst-case error in estimating the result size of the range-selection query, $\sigma_{X_\alpha \le X \le X_\beta}(R)$, where the attribute values X_α and X_β fall completely within the k^{th} T-ACM sector is given by,*

$$\varepsilon = \begin{cases} n_k - \mathcal{A} & \text{if } \mathcal{A} < \frac{n_k}{2}, \\ \mathcal{A} & \text{if } \mathcal{A} \ge \frac{n_k}{2} \end{cases}$$

where $\mathcal{A}$ is the expected number of tuples between the attribute values X_α and X_β and is equal to,

$$\mathcal{A} = a_k(\beta - \alpha + 1) + \frac{(n_k - a_k l)(\beta - \alpha + 1)(\beta - \alpha + 2)}{l(l-1)}.$$ □

Theorem 9. *Assuming that the T-ACM sector has been obtained by processing a histogram bucket of size l with n_j tuples, the average error in estimating the result size of the equality selection query, $\sigma_{X=X_i}(R)$, obtained by averaging over* all *attribute values in this sector of the trapezoidal ACM is exactly zero.* □

Theorem 10. *The upper bound of the average-case error, ε, in estimating the result size of an equality select query, $\sigma_{X=X_i}(R)$, using a trapezoidal ACM is,*

$$\varepsilon = a_k + \frac{2(n_k - a_k l)}{l(l-1)}.i - \frac{n_k}{l},$$

where a_k is the frequency of the first attribute value in the k^{th} sector and X_i is in the i^{th} position of the T-ACM sector. □

Theorem 11. *The average-case error in estimating the result size of the selection query, $\sigma_{X_\alpha \le X \le X_\beta}(R)$, where the attribute values X_α and X_β fall completely within the k^{th} T-ACM sector is given by,*

$$\varepsilon = \frac{(\beta - \alpha)(\alpha + \beta - 3)(n_k - a_k l)}{l - 1}$$

where $\beta > \alpha$. □

Lemma 18. *Considering the equality join of two domain compatible attributes X and Y with $X_i = Y_j$, if the expected result size of the equality selection query, $\sigma_{X=X_i}$, using an ACM is $\hat{x}_i$ and that of $\sigma_{Y=Y_j}$ is $\hat{y}_j$, then the maximum error resulting from joining the attributes X and Y on the values X_i and Y_j is given by,*

$$\varepsilon = |(\hat{x}_i \varepsilon_y + \hat{y}_j \varepsilon_x + \varepsilon_x \varepsilon_y)|$$

where ε_x and ε_y are the estimated errors resulting from the equality selection queries $\sigma_{X=X_i}$ and $\sigma_{X=Y_j}$ respectively. □

Corollary 1. *The error resulting from an equality join of two domain compatible attributes X and Y, is given by,*

$$\varepsilon = \sum_{j=1}^{s_X} \sum_{i=1}^{l_j} (\hat{x}_i \varepsilon_{y_k} + \hat{y}_k \varepsilon_{x_i} + \varepsilon_{x_i} \varepsilon_{y_k})$$

where k is an index on the T-ACM of Y such that $X_i = Y_k$ and $\varepsilon_{x_i}, \varepsilon_{y_k}$ are the errors resulting from the equality selection queries $\sigma_{X=X_i}$ and $\sigma_{Y=Y_k}$ respectively. □

Again, more properties of the T-ACM are described in [15] and [1]. Its power in minimizing error estimates for query result sizes, and in determining nearly-optimal QEPs for theoretical databases, is found in [15].

The T-ACM, in its purest form, is not so powerful except if the form of the histogram distribution is truly (or approximately) linear [1]. To overcome this inherent handicap, in [21], we introduced a modified version called the Bounded T-ACM (or the BT-ACM). This is an improved histogram based on the T-ACM, which possesses the additional property that the error of the estimate is *bounded* by a user-specified quantity from the linear approximation dictated by the T-ACM.

Both the R-ACM and the T-ACM can be used as either a univariate or a multivariate method. Throughout this paper, we consider only the univariate application. We believe that the results hold for multivariate distributions as well.

2.2 QEP Determinations in ORACLE

We *informally* describe[4] the process used in query optimization below. In any query optimizer (and in particular, in the ORACLE DBMS), queries are evaluated, and a hopefully optimal QEP is chosen after the following procedure:

(i) The optimizer checks the available access paths for the given query, and decides on a set of potential candidate QEPs.
(ii) The optimizer searches in the space of potential QEPs, and estimates the costs for all potential candidate QEPs by using the statistical information and cost models provided by the DBMS.
(iii) The optimizer chooses the one with the least cost as the "best" QEP for the given query.

It is clear that the computation in Step (ii) above requires various pieces of information used to determine the quality of any QEP. Typical pieces of useful information involve statistical information, the respective selectivity indices, and the cost model. Each of these are briefly described below.

[4] Clearly, this is a *very* informal description. Detailed descriptions about how this is achieved can be found in [22] and [3], and in the standard textbooks and product manuals. However, it is worth mentioning, that the process that we are concerned with deals with the use of the *histograms* in determining the best QEP, namely by invoking the corresponding histogram method to estimate the query result sizes. Indeed, we intend to merely "substitute" the currently-invoked histogram method by an ACM.

The statistics used by the query optimizer contains information about the features of the data and its distribution, which are, in turn, stored by the DBMS in terms of data dictionaries or catalogues. Such pieces of statistical information form the foundational basis using which the query optimizer evaluates candidate QEPs.

The term "selectivity" refers to the fraction of rows to be selected from a row set. It is associated with a query predicate. A predicate acts as a filter that filters a certain number of rows which do not meet the specified condition. Therefore, the selectivity of a *predicate* indicates the number of rows which pass the predicate filter. A selectivity of zero means that no rows will be selected from a row set, and a selectivity of unity means that all the rows will be selected. The cost represents the units of resources used for answering the query. Generally, three categories of resources are taken into account, namely disk I/O accesses, CPU usage, and memory usage.

3 Integration of the Methods into Oracle

The database system we chose to work with is the ORACLE DBMS, whose version is the ORACLE 9*i* Enterprise Edition for the Windows XP operating system. Our goal was to integrate the ACM histogram-based algorithms, into the existing database query optimizer to serve the purpose of a "plug-in". Since the ORACLE DBMS is one of the most widely used commercial database systems, we believe that this renders our experimental results both meaningful and extremely pertinent to the real-world application domain. Furthermore, the numerous development tools supplied by the ORACLE system provide great flexibility for application development, and make the integration feasible.

The goal of our implementation is that the ORACLE query optimizer should be capable of utilizing the underlying ACM methods in approximating query result sizes, and in determining QEPs when evaluating and optimizing queries sent to the DBMS. Thus, our implementation should possess two fundamental characteristics. First of all, invoking and utilizing the new histogram methods for arbitrary queries should be simple and easy to use. This would render the complicated procedures involved with invocation and computation completely transparent to users. Secondly, the implementation should fully integrate the new ACM histograms into the ORACLE query optimizer, so that all the computations for the selectivity and cost are automatically computed as per the new histogram method. Indeed, we have successfully achieved these requirements (at least from a prototype prima-facie point of view), and based on these achievements, we believe that our prototype implementation can well serve the task of being a "plug-in" to the ORACLE DBMS. While the implementation details are omitted, the salient features of this "plug-in" are listed below.

3.1 The Core Architecture of Our Implementation

The architecture of the implementation (See Figure 1) is composed of three major components: (i) the operators, (ii) the catalogue, and (iii) the algorithms being incorporated. All these three components are internal to the query optimizer, and furthermore, the components (i) and (iii) are inside the *estimator* routine itself. In other words, we have

attempted to extend the query optimizer by utilizing the new histogram-based algorithms, and in turn, by computing the corresponding estimates using these *new* algorithms. These three components work seamlessly with the ORACLE query optimizer. Indeed, the interface of the "plug-in" interacts directly with the system's query optimizer. This implies that the input to the "plug-in" comes from the query optimizer, and the output goes to the query optimizer too. Thus, the rest of computations involved in the QEP determination remain unchanged. It should thus be emphasized that *these computations* are achieved automatically by the query optimizer using ORACLE's internally designed methods. A brief description about how this is achieved follows.

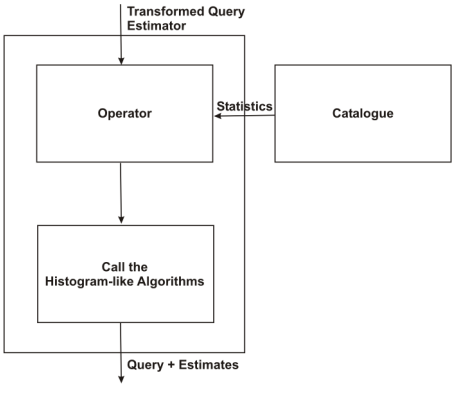


Fig. 1. The Architecture of our implementation

• **Catalogues**
Special catalogues are created in the database system in order to store the statistical information required for the R-ACM and T-ACM histograms. The functions of these specifically created catalogues are identical to the other catalogues or relations stored in the ORACLE database system for their inherent histograms, except that their specific characteristic lies in the fact that they are used for the R-ACM and T-ACM histograms. Therefore, these catalogues can be stored either in a user's private space, or in a public space that is accessible for all the database users. Populating these catalogues is accomplished by invoking ORACLE's standard modules/commands that are used for collecting statistical information. Deleting the content of these catalogues can be done in an analogous manner.

• **Operators**
The term "Operator" is used in ORACLE to allow the query optimizer to compute user-defined selectivity and cost, and to collect and delete user-defined statistical information. The operator can be treated as an extension of the SQL syntax, and can be used in any standard SQL query, whenever it is needed. In our prototype, we developed two operators specifically for our present "plug-in" and for our experimental set-up. These two operators were called the RACM and the TACM respectively, which represent the process of utilizing (or invoking) the R-ACM and the T-ACM algorithms respectively.

Following exactly the same methods of computation used by ORACLE, these operators achieve the major functions listed below:

(i) **Collecting and deleting corresponding statistical information**
Whenever the standard command for collecting or deleting statistics is invoked, the corresponding statistical information for the R-ACM (or the T-ACM) is accordingly collected or deleted for the underlying attribute of the corresponding relation. In other words, this process has been rendered automatic so that it, inherently, is associated with the entire DBMS.

(ii) **Computation of the selectivity**
Whenever an operator is invoked in an SQL statement, the selectivity for that particular predicate is computed by using the corresponding algorithm, namely, either the R-ACM or the T-ACM, in the experiments.

3.2 Selectivity Computation for Queries

In this paper, since we are dealing with a prototype *prima-facie* case "plug-in", we mainly focus on queries belonging to two categories, namely the equi-selections and the equi-joins. For the former category, the accuracy of the estimates of the query result sizes directly affects the ultimate estimates of the queries. For the latter category, the accuracy plays an important role in also choosing the join orders and join methods, as well as in affecting the ultimate estimates. All the experimental results reported in this paper are based on these two categories of queries, and the results are listed in the next section.

4 Experimental Results

4.1 Test Data Sets

The objective for our experiments is to test and present the performance of the R-ACM and T-ACM on a real database system. Also, we want the data sets used in the test, to possess real-life (or close to real-life) characteristics, so that the test results are more "accountable" and trustworthy. Based on the above considerations, we chose to utilize the ORACLE system on the TPC-H data sets.

The TPC-H benchmark is a decision support benchmark designed and implemented by the Transaction Processing Performance Council (TPC) organization. It is a standardized benchmark in the broader database industry. It consists of a data generation tool, a suite of business oriented *ad-hoc* queries, and concurrent data modifications. Both the queries, and the data comprising the database have been chosen as an industry-wide pertinent standard. This benchmark illustrates most of the major characteristics required of decision support systems, which involve examining large volumes of data, executing queries with a high degree of complexity, and giving answers to critical business questions. It thus reflects the real-life business requirements of an enterprise in an elegant and complete manner.

4.2 Sample Queries

In our entire experiments, we examined twelve sample queries specified by the TPC-H database. These are, indeed, *all* the queries which deal with our domain of interest, namely those involving equi-select and equi-join. Since we are only interested in a *prim-facie* case (as opposed to a commercial product), the queries we considered are simplified versions derived from the standard TPC-H benchmark queries. But we add that although the queries were simplified, their essential characteristics that necessitated the histogram invocations, were preserved. Among these twelve, four sample queries constituted simple equi-selections, and the rest were equi-join queries. The number of relations involved in the joins was as large as six. The details of the sample queries can be gleaned from the TPC-H specifications, and are included in [3].

Table 2. Queries Q2, Q3, Q5, Q7, and Q9 from the TPC-H Database used in our experiments

Query No.	Query	Description
Q2	*SELECT* $*$ *FROM PART, SUPPLIER, PARTSUPP,* *NATION, REGION* *WHERE R_REGIONKEY* = 3 *AND N_REGIONKEY= R_REGIONKEY* *AND S_NATIONKEY = N_NATIONKEY* *AND P_SIZE* = 20 *AND PS_PARTKEY = P_PARTKEY* *AND PS_SUPPKEY = S_SUPPKEY*;	This query finds which supplier should be selected to place an order for a given part in a given region.
Q3	*SELECT* $*$ *FROM LINEITEM, ORDERS, CUSTOMER* *WHERE L_ORDERKEY = O_ORDERKEY AND* *C_CUSTKEY = O_CUSTKEY* *AND O_CUSTKEY* = 800;	This query retrieves the information about given customer.
Q5	*SELECT* $*$ *FROM ORDERS, LINEITEM, CUSTOMER,* *SUPPLIER, NATION, REGION* *WHERE* O_O*RDERKEY = L_ORDERKEY* *AND L_SUPPKEY = S_SUPPKEY* *AND C_CUSTKEY = O_CUSTKEY* *AND C_NATIONKEY = S_NATIONKEY* *AND S_NATIONKEY = N_NATIONKEY* *AND N_REGIONKEY = R_REGIONKEY* *AND S_NATIONKEY* =8;	This query lists information about suppliers.
Q6	*SELECT* $*$ *FROM LINEITEM WHERE* *L_QUANTITY* = 25;	This query gives *LINEITEM* information.
Q7	*SELECT* $*$ *FROM LINEITEM, ORDERS, CUSTOMER,* *SUPPLIER, NATION* *WHERE* S_S*UPPKEY = L_SUPPKEY* *AND L_ORDERKEY = O_ORDERKEY* *AND C_CUSTKEY = O_CUSTKEY* *AND C_NATIONKEY = S_NATIONKEY* *AND N_NATIONKEY = S_NATIONKEY* *AND S_NATIONKEY* = 3;	This query determines the shipping information.
Q9	*SELECT* $*$ *FROM SUPPLIER, PARTSUPP, LINEITEM,* *ORDERS, NATION* *WHERE PS_SUPPKEY = L_SUPPKEY* *AND L_ORDERKEY = O_ORDERKEY* *AND S_SUPPKEY = L_SUPPKEY* *AND S_NATIONKEY = N_NATIONKEY* *AND S_NATIONKEY* =3;	This query determines the information on lines of parts.

Table 3. Queries Q10, Q11, Q12, Q18', Q19' and Q20' from the TPC-H Database used in our experiments

Query No.	Query	Description
Q10	*SELECT* * *FROM LINEITEM, ORDERS, CUSTOMER, NATION* *WHERE C_CUSTKEY = O_CUSTKEY* *AND L_ORDERKEY = O_ORDERKEY* *AND C_NATIONKEY = N_NATIONKEY* *AND O_CUSTKEY* = 5000;	This query identifies customers who have parts shipped to them.
Q11	*SELECT* * *FROM PARTSUPP, SUPPLIER, NATION* *WHERE PS_SUPPKEY = S_SUPPKEY* *AND S_NATIONKEY = N_NATIONKEY* *AND S_NATIONKEY* = 3;	This query finds the suppliers' stock in a given nation.
Q12	*SELECT* * *FROM ORDERS, LINEITEM* *WHERE O_ORDERKEY = L_ORDERKEY* *AND L_QUANTITY* = 25;	This query determines the relationship between the shipping modes and priority orders.
Q18'	*SELECT* * *FROM PART* *WHEREP_SIZE* =20;	This query lists parts information with certain size.
Q19'	*SELECT* * *FROM CUSTOMER* *WHERE C_NATIONKEY* = 10;	This query lists information about customers in given nation.
Q20'	*SELECT* * *FROM SUPPLIER* *WHERE S_NATIONKEY*=3;	This query lists supplier information for a given nation.

The experiments conducted in this study were executed so as to compare the performance of different histogram-based algorithms in estimating query result sizes *and* in the QEP selection. Thus, there are two sets of comparisons, namely those comparing the R-ACM with the Equi-depth, and those comparing the BT-ACM with the Equi-depth. Also, the comparisons were made with two different kinds of data distributions, namely the uniform and the multi-fractal distributions [3].

The experiments conducted in this study were executed so as to compare the performance of different histogram-based algorithms in estimating query result sizes and in the QEP selection. Thus, there are two sets of comparisons, namely those comparing the R-ACM with the Equi-depth, and those comparing the BT-ACM with the Equi-depth. Also, the comparisons were made with two different kinds of data distributions, namely the uniform and the multi-fractal distributions [3].

However, the actual "simplified" queries used in the testing are detailed below in Tables 2 and 3.

4.3 Accuracy of the Error Estimates

Comparing the R-ACM vs. Equi-depth. For each set of experiments, all the sample queries were tested against both the R-ACM and the Equi-Depth schemes. For the R-ACM experiments, we resorted to two different storage requirement options, namely the option of using the same number of buckets, and the option of using the same storage

Table 4. A comparison of the estimated query result sizes obtained by using the R-ACM and Equi-depth for uniform distributions. The $R-ACM1$ is the R-ACM with same number of buckets, and the $R-ACM2$ is the R-ACM with same storage requirements.

Query	Act. size	Estim. size			Error Rate (%)		
		$R-ACM1$	$R-ACM2$	$Equi-depth$	$R-ACM1$	$R-ACM2$	$Equi-depth$
Q2	1,533	1,538	1,597	1,601	0.326158	4.174821	4.435747
Q3	30	40	40	71	33.33333	33.33333	136.66667
Q5	5,203	5,135	4,967	7,220	1.306938	4.535845	38.766111
Q6	60,251	60,222	60,282	59,994	0.048132	0.051451	0.426549
Q7	5,190	5,021	4,967	7,220	3.256262	4.296724	39.113681
Q9	10,111,600	10,042,899	9,934,910	9,624,692	0.679428	1.747399	4.815341
Q10	39	41	40	71	5.128205	2.564103	82.051282
Q11	16,800	16,740	16,560	16,043	0.357143	1.428571	4.505952
Q12	60,251	60,222	60,282	59,994	0.048132	0.051451	0.426549
Q18'	1,997	1,997	1,997	2,001	0	0	0.200300
Q19'	2,961	2,961	2,980	3,002	0	0.641675	1.384667

Table 5. A comparison of the estimated query result sizes obtained by using the R-ACM and Equi-depth for Multi-Fractal Distributions (p=0.2). The $R-ACM1$ is the R-ACM with same number of buckets, and the $R-ACM2$ is the R-ACM with same storage requirements.

Query	Act. size	Estim. size			Error Rate (%)		
		$R-ACM1$	$R-ACM2$	$Equi-depth$	$R-ACM1$	$R-ACM2$	$Equi-depth$
Q2	1256	1282	1390	1098	2.070	10.6688	12.580
Q3	3702	43227	34671	104881	1067.666	836.5478	2733.090
Q5	71	977	365	5669	1276.056	414.0845	7884.507
Q6	196585	196585	196585	187479	0	0	4.632
Q7	13	45	87	5669	246.154	569.2308	43507.690
Q9	1519760	920629	2523616	12970317	39.423	66.0536	753.445
Q10	3702	42498	34087	104881	1047.974	820.7726	2733.090
Q11	2560	1535	4206	21620	40.0391	64.2969	744.531
Q12	196585	196585	196585	187449	0	0	4.647
Q18'	1638	1638	1775	1402	0	8.3639	14.408
Q19'	1536	1536	1232	3727	0	19.7917	142.643

requirements. The formula used to compute the storage requirements is : $s_{(Equi-depth)} = 2 * s_{(R-ACM)} - 1$, where s is the number of buckets. With respect to the actual histograms built for the attributes in the sample queries, a few issues have to be clarified. First of all, since the R-ACM and the BT-ACM were integrated into the ORACLE query optimizer by using their corresponding "operators", they were capable of performing *all* the same functions which ORACLE's default histogram algorithm, the Equi-depth, could. The results obtained are given in Tables 4 and 5. From the tables, it is obvious that the R-ACM yielded superior performance over the Equi-depth for all the queries, and with any storage requirements considerations, whether the R-ACM had the same number of buckets or the same storage requirements. Also noticeable is the outstanding performance of the R-ACM for the multi-fractal distributions (See Tables 4 and 5). For example, in Table 4 when we consider the option of using the same storage, the error rates of sample queries $Q3$, $Q5$, $Q6$, $Q10$, $Q11$ and $Q18'$ for the R-ACM are significantly less than those for the Equi-depth. More specifically, for the sample query Q10 which involves a four relation-join, the error rate for the R-ACM, which is

approximately 2.56%, is nearly 40 times less than that for the Equi-depth, which is approximately 82.05%. Similarly, consider Table 5 where we analyze the option of using the same storage. In this case, the sample query $Q7$ involves five relations in the underlying join. The error rate of this query for the R-ACM, which is approximately 569.23%, is nearly 80 times less than that for the Equi-depth, which is approximately 43507.69%. These results and the tables in their entirety, prove the superiority of the R-ACM on estimating query result sizes. The reader should also observe that the concern of increasing the storage requirements for the R-ACM is really, not a valid concern.

4.4 Comparing the BT-ACM and Equi-depth for Uniform Distributions

It was argued in [2] that the T-ACM could not yield superior performance on non-linear distributions. As an enhancement of the T-ACM, the BT-ACM introduced in [3] maintains the most significant feature of the T-ACM, which is, that each bucket has a trapezoidal shape. It is obvious that such a distribution will not necessarily yield a superior performance on the multi-fractal distributions, since the multi-fractal distributions are the most extreme cases of non-uniform distributions. For these reasons, the BT-ACM histograms were tested for only uniform distributions. Different from the tests conducted in the previous set of experiments, we chose four equi-selection queries, and tested all the possible values for the underlying attributes. The experimental results are shown in Table 6.

Table 6. Experimental results comparing the BT-ACM and Equi-depth histograms for uniform distributions

Query	No. of Better Estimator		Overall Estim. Error Rate (%)	
	BT-ACM	Equi-depth	BT-ACM	Equi-depth
Q6	38	12	0.120071	0.3354168
Q18	41.5	8.5	0.618160	2.1322956
Q19'	19	6	0.51595	1.2104582
Q20'	21	4	1.429133	4.4767138

For the BT-ACM experiments, we observe in the first place, that the number of buckets is far less than those used for the Equi-depth algorithm. Therefore we can effectively consider both of these two algorithms to be having the same storage requirements. Note that during the experiments, a tolerance value, which is similar to the one used in the R-ACM, was carefully chosen to yield more accurate estimates, and to control the number of generated buckets. From Table 6, we can see that the number of cases when the BT-ACM yields better estimates is much more than when the Equi-depth is superior. Besides, the BT-ACM yields superior overall performance. We are thus fairly confident that we can effectively utilize the BT-ACM in approximating frequency distributions for these database applications.

4.5 QEP Selection Comparison

As mentioned earlier, the major contribution of this research endeavor is to demonstrate that better QEPs are chosen by utilizing the R-ACM in the ORACLE query optimizer.

To demonstrate this we present below more details of our "plug-in" implementation. First of all, we note that in the entire implementation, we did not modify or change the fundamental rules of how the ORACLE query optimizer selects the "best" QEP. As described earlier, what we have implemented can be considered as an extension to the original ORACLE query optimizer, which allows it to utilize external histogram algorithms to achieve the task of estimating query result sizes and choosing QEPs. Thus, in the implementation, we did not take into account the issues involving QEP search-space pruning and QEP selection-rule optimization. Instead, these are accomplished by the ORACLE query optimizer following the original inherent design and execution. Since the tolerance value, τ, was appropriately chosen, the storage requirements for the R-ACM and the Equi-depth can be effectively considered to be the same, and hence the cost for the operations in the QEPs involving these algorithms can therefore be also considered to be identical. As a result of this, we can now compare these two algorithms by comparing the *final costs* of the QEPs that they respectively choose. The accuracy of the selectivity, which is determined by the estimate of the query result sizes, is therefore the most critical factor in selecting the optimal QEP. In this connection we observe that with respect to the join orders for join operations, it is also true that the R-ACM seems to always choose superior solutions. To illustrate this, we would like to take a few examples demonstrating the respective QEPs chosen by using these two algorithms. While it is certainly difficult to explain this and present the trace for a *large* number of cases, we shall do it for representative cases for *all the TPC-H queries* under consideration. In each case, we list the SQL statements for the queries, and the screenshots for the chosen QEPs. Rather than give a detailed analysis of all the cases (which can be quite repetitive), we shall do this with an analysis of two particular scenarios, namely, for the query in Example 1, and Query 5 (in Example 2). The rest of the cases are included in the Appendix, but without detailed explanations.

4.6 Example 1

Consider the query below where the results given are obtained from the uniform distribution.

$$
\begin{aligned}
&SELECT * \\
&FROM\ SUPPLIER, CUSTOMER, \\
&ORDERS, PARTSUPP \\
&WHERE \\
&S\_NATIONKEY = C\_NATIONKEY \\
&AND\ O\_CUSTKEY = C\_CUSTKEY \\
&AND\ S\_SUPPKEY = PS\_SUPPKEY \\
&AND\ S\_NATIONKEY = 22;
\end{aligned}
$$

The cost for each step of the operations in the QEPs is shown in parenthesis in Figure 2. Note that the join methods chosen by the R-ACM for each join operation is the hash join, which is the most efficient method when a join operation returns a large

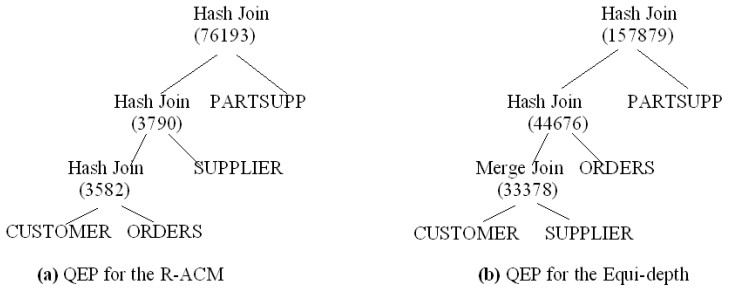


Fig. 2. The QEPs Chosen for the Sample Query of Example 1

number of tuples. Subsequently, the cost in (b), for every step after the first step of the join (which is represented at the bottom of the operation tree), is much greater than the corresponding step given in (a). Furthermore, the final cost for the Equi-depth's QEP, 157,879, is significantly larger than that for the R-ACM's QEP, namely, 76,193, which is shown at the top of each operation tree.

4.7 Example 2

We consider the case when the QEP is determined for the TPC-H Query Q5 by using the R-ACM and the Equi-depth histograms respectively, based on using the same storage configurations. The results showed that all the QEPs chosen for the R-ACM are better than those chosen for the Equi-depth. This is true for criteria measured in terms of both the cost and the efficiency. The query are given below; the the corresponding screen-shots are given in Figure 3.

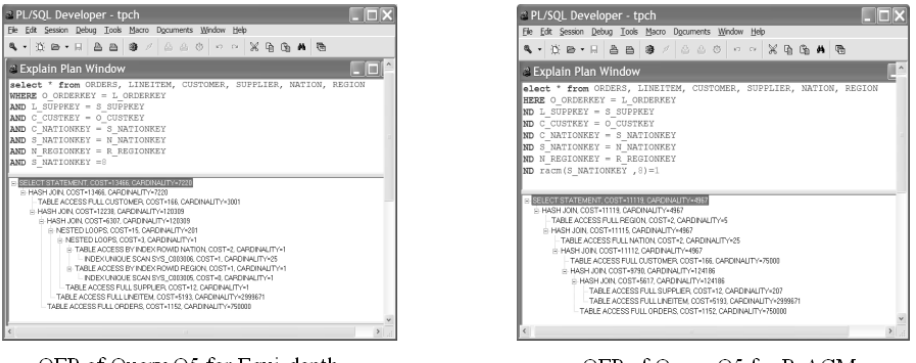


Fig. 3. The QEPs Chosen for the Sample Query 5

SQL Statement for Q5
$SELECT * FROM$
$ORDERS, LINEITEM, CUSTOMER,$
$SUPPLIER, NATION, REGION$
$WHERE$
$O\_ORDERKEY = L\_ORDERKEY$
$AND\ L\_SUPPKEY = S\_SUPPKEY$
$AND\ C\_CUSTKEY = O\_CUSTKEY$
$AND\ C\_NATIONKEY = S\_NATIONKEY$
$AND\ S\_NATIONKEY = N\_NATIONKEY$
$AND\ N\_REGIONKEY = R\_REGIONKEY$
$AND\ S\_NATIONKEY = 8;$

Actual result size: $5,203$
Cost of the QEP chosen by the R-ACM: $11,119$
Cost of the QEP chosen by the Equi-Depth: $13,466$.

The traces of the QEPs selected for Query 5 are shown in Figure 3 for the Equi-depth and the R-ACM algorithms respectively. This is the query which involves the most relations from the entire set of sample queries. The ultimate cost of the QEP for the R-ACM was $11,119$, while that for the Equi-depth was $13,466$. The final query result size of this sample query made by the R-ACM was $4,967$, and the corresponding quantity for the Equi-depth was $7,220$. Notice that the real result size is $5,203$. It is quite easy to see that the R-ACM leads to a lower cost, and to a more accurate estimation of the ultimate query result sizes than the Equi-depth. Also, we can conclude that the lower costs result purely from the more accurate estimates, because the selectivity criterion is the only issue that we have considered when we attempted to obtain improvements for the QEP selection process. Clearly, the performance of the QEP is also improved by this enhancement.

Table 7. Experimental results comparing the R-ACM and Equi-depth Histograms: Cost and final result size

Query	Final Result Size estimated by the Equi-depth	Cost using Equi-dept	Final Result Size estimated by the R-ACM	Cost using R-ACM
Q2	1601	2039	1597	2039
Q3	71	1190	40	1182
Q5	7220	13,466	4967	11,119
Q6	60,282	5193	60,222	5193
Q7	7220	12,030	4967	11,115
Q9	9,624,692	14,856	9,934,910	13,718

The comparison of the Equi-depth and R-ACM for all the relevant queries is given in Table 7. The advantages of the R-ACM is obvious. For example, for query Q5, the cost of using the R-ACM is $11,119$ compared to $13,466$ required for the Equi-depth, and the final result sizes is $4,967$ for the R-ACM as opposed to $7,220$ for the Equi-depth.

5 Conclusions

In this paper, we have studied the fundamental problem encountered in database query optimization, namely, the problem of estimating query result sizes, and the question of determining the QEP prior to executing the query. Typically, this is achieved by using histograms in approximating query result sizes in the query optimizer. In particular, we have attempted to study the applicability of the R-ACM and a variant of the T-ACM introduced by Oommen and Thiyagarajah [1], [14] and [15]. Although the superiority of these in yielding more accurate query result size estimates was earlier clearly demonstrated [13] and the resulting superior QEPs for a theoretically-modeled database was shown in [2], in this paper we demonstrated how the ACMs could be incorporated into a *real-life* DBMS. This was done by designing and implementing a prototype which sits on top of an ORACLE 9*i* system. The integration was achieved in C/C++ and PL/SQL, and serves as a prototype "plug-in" to the ORACLE system, since it is fully integrated and completely transparent to users. The superiority of utilizing the ACM histograms was rigorously validated by conducting an extensive set of experiments on the TPC-H benchmark data sets, and by testing on equi-select and equi-join queries. The future work consists of developing a full-fledged application for ORACLE and other DBMSs, and hopefully, marketing them successfully.

References

1. Thiyagarajah, M.: Attribute Cardinality Maps: New Query Result- Size Estimation Techniques for Database Systems. Ph.D. Thesis, School of Computer Science, Carleton Univ., Ottawa (1999)
2. Oommen, B.J., Rueda, L.G.: The efficiency of histogram-like techniques for database query optimization. The Computer Journal 45, 494–510 (2002)
3. Chen, J.: On Utilizing New Histogram-Based Methods for Query Optimization. MCS. Thesis, School of Computer Science, Carleton Univ., Ottawa, Canada (2003)
4. Ioannidis, Y., Poosala, V.: Histogram-based solutions to diverse database estimation problems. IEEE Data Engineering 18, 10–18 (1995)
5. Ioannidis, Y., Poosala, V.: Balancing histogram optimality and practicality for query result size estimation. In: Proc. of the ACM SIGMOD Conference, pp. 233–244 (1995)
6. Ioannidis, Y., Poosala, V.: Histogram-based approximation of set-valued query-answers. In: Proc. of the VLDB Conference, vol. 8, pp. 174–185 (1999)
7. Poosala, V., Ioannidis, Y., Haas, P., Shekita, E.: Improved histograms for selectivity estimation of range predicates. In: Proc. of the ASM SIGMOD Conference, pp. 294–305 (1996)
8. Vitter, J.S., Wang, M.: Approximate computation of multidimensional aggregates of sparse data using wavelets. In: Proc. of the ACM SIGMOD Conference, pp. 193–204 (1999)
9. Thaper, N., Guha, S., Indyk, P., Koudas, N.: Dynamic multidimensional histograms. In: Proc. of the ACM SIGMOD Conference, pp. 428–349 (2002)
10. Haas, P.J., Swami, A.N.: Sequential sampling procedures for query size estimation. In: Proc. of the ACM SIGMOD Conference, pp. 341–350 (1992)
11. Haas, P.J., Naughton, J.F., Seshadri, S., Stokes, L.: Sampling-based estimation of the number of distinct values of an attribute. In: Proc. of the VLDB Conference, pp. 311–322 (1995)
12. Oommen, B.J., Chen, J.: A new histogram method for sparse attributes: The averaged rectangular attribute cardinality map. In: Proceedings of ISICT 2003, the 2003 International Symposium on Informtaion and Communication Technologies, Ireland, pp. 119–125 (2003)

13. Oommen, B.J., Thiyagarajah, M.: Benchmarking attribute cardinality maps for database systems using the TPC-D specifications. IEEE Transactions on Systems, Man and Cybernetics SMC-33(B), 913–924 (2003)
14. Oommen, B.J., Thiyagarajah, M.: Rectangular attribute cardinality map: A new histogram-like technique for query optimization. In: Proc. Of the International Database Engineering and Applications Symposium, pp. 3–15 (1999)
15. Oommen, B.J., Thiyagarajah, M.: Query result size estimation using the trapezoidal attribute cardinality map. In: Proc. of the International Database Engineering and Applications Symposium, pp. 236–242 (2000)
16. Ioannidis, Y., Christodoulakis, S.: Optimal histograms for limiting worst-case error propagation in the size of join results. ACM Transactions on Database Systems (TODS) 18, 709–748 (1993)
17. Piatetsky-Shapiro, G., Connell, C.: Accurate estimation of the number of tuples satisfying a condition. In: Proc. of the ACM SIGMOD Conference, pp. 256–276 (1984)
18. Mannino, M.V., Chu, P., Sager, T.: Statistical profile estimation in database systems. ACM Computing Surveys 20, 191–221 (1988)
19. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, Boston (1990)
20. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley, New York (2000)
21. Oommen, B.J., Chen, J.: The bounded trapezoidal attribute cardinality map and its application to query optimization. In: Proceedings of ICCSI 2003, the 2003 International Conference on Computer Science and Informatics, North Carolina, USA, pp. 422–426 (2003)
22. Jarke, M., Koch, J.: Query optimization in database systems. ACM Computing Surveys 16, 311–322 (1984)

Appendix

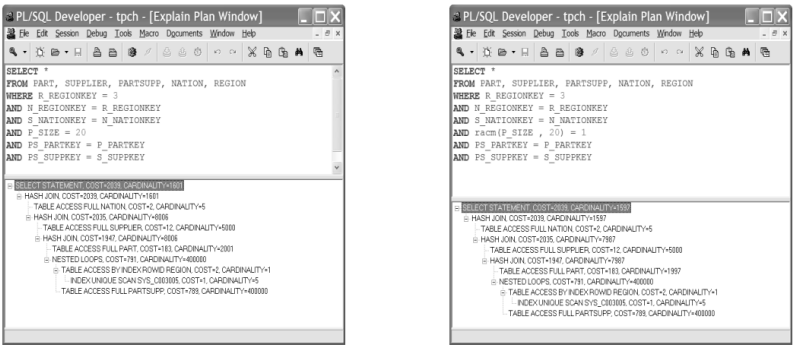
In this Appendix we give a complete listing of the screen-shots obtained from our "plug-in" when the various queries of the TPC-H benchmark database and queries are tested using the R-ACM and the Equi-depth histograms. As mentioned in the body of the paper, the results (presented below) showed that almost all the QEPs chosen for the R-ACM are better than those chosen for the Equi-depth, and this is true for criteria measured in terms of both the cost and the efficiency. In each case we list below the SQL statements for the queries, and the screenshots for the chosen QEPs. Since the figures are self-explanatory, we omit a detailed analysis of all the cases, which can be quite repetitive.

1. Example A.1 : SQL Query for Q2.

$$
\begin{aligned}
&SELECT * \\
&FROM\ PART, SUPPLIER, PARTSUPP, NATION, REGION \\
&WHERE\ R\_REGIONKEY = 3 \\
&AND\ N\_REGIONKEY = R\_REGIONKEY \\
&AND\ S\_NATIONKEY = N\_NATIONKEY \\
&AND\ P\_SIZE = 20 \\
&AND\ PS\_PARTKEY = P\_PARTKEY \\
&AND\ PS\_SUPPKEY = S\_SUPPKEY;
\end{aligned}
$$

Actual result size: 1,533.
Cost of QEP determined by the R-ACM: 2039.
Final result size estimated by the R-ACM: 1597.
Cost of QEP determined by the Equi-depth: 2039.
Final result size estimated by the Equi-depth: 1601.

The corresponding screenshots for the two QEPS chosen for Q2 are given in Figure 4.



QEP of Query Q2 for Equi-depth

QEP of Query Q2 for R-ACM

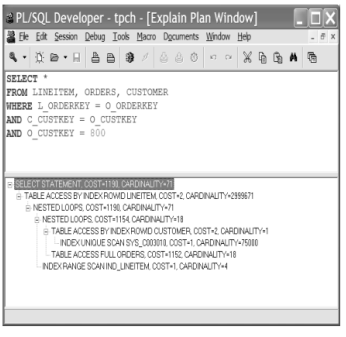
Fig. 4. The QEPs Chosen for the Sample Query 2

2. Example 5.2.b : SQL Query for Q3.
   SELECT ∗
   FROM LINEITEM, *ORDERS*, *CUSTOMER*
   WHERE L_ORDERKEY = *O_ORDERKEY*
   AND C_CUSTKEY = *O_CUSTKEY*
   AND O_CUSTKEY = 800;
   Actual result size: 30.
   Cost of QEP determined by the R-ACM: 1182.
   Final result size estimated by the R-ACM: 40.
   Cost of QEP determined by the Equi-depth: 1190.
   Final result size estimated by the Equi-depth: 71.

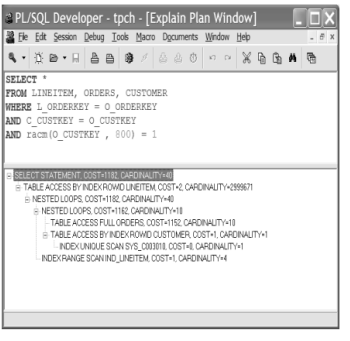
   The corresponding screenshots the two QEPS chosen for Q3 are given in Figure 5.

3. Example 5.2.d : SQL Query for Q6.
   SELECT ∗
   FROM LINEITEM WHERE L_QUANTITY = 25;

Actual result size: 60,251.
Cost of QEP determined by the R-ACM: 5193.
Final result size estimated by the R-ACM: 60,222.
Cost of QEP determined by the Equi-depth: 5193.
Final result size estimated by the Equi-depth: 60,282.



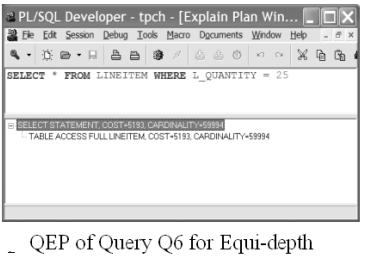
QEP of Query Q3 for Equi-depth



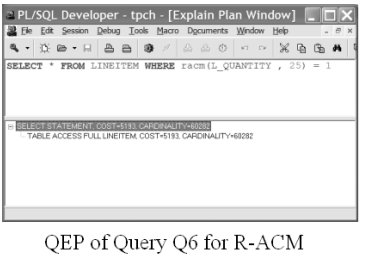
QEP of Query Q3 for R-ACM

Fig. 5. The QEPs Chosen for the Sample Query 3

The corresponding screenshots for the two QEPS chosen for Q6 are given in Figure 6.



QEP of Query Q6 for Equi-depth



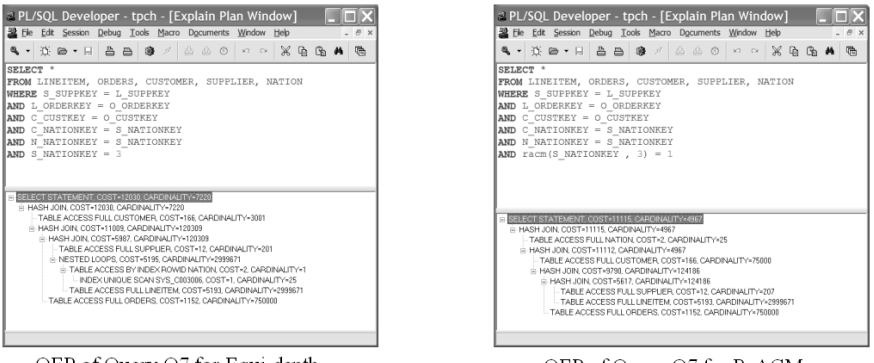
QEP of Query Q6 for R-ACM

Fig. 6. The QEPs Chosen for the Sample Query 6

4. Example 5.2.e : SQL Query for Q7.
   SELECT $*$
   FROM LINEITEM, ORDERS, CUSTOMER, SUPPLIER, NATION
   WHERE S_S*UPPKEY = L_SUPPKEY*
   AND L_ORDERKEY = O_ORDERKEY
   AND C_CUSTKEY =O_CUSTKEY
   AND C_NATIONKEY = S_NATIONKEY

AND N_NATIONKEY =S_NATIONKEY
AND S_NATIONKEY = 3;
Actual result size: 5,190.
Cost of QEP determined by the R-ACM: 11115.
Final result size estimated by the R-ACM: 4967.
Cost of QEP determined by the Equi-depth: 12030.
Final result size estimated by the Equi-depth: 7220.

The corresponding screenshots for the two QEPS chosen for Q7 are given in Figure 7.



QEP of Query Q7 for Equi-depth

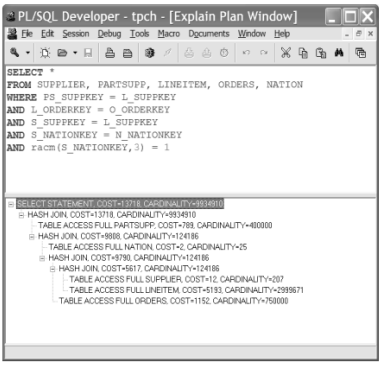
QEP of Query Q7 for R-ACM

Fig. 7. The QEPs Chosen for the Sample Query 7

5. Example 5.2.f : SQL Query for Q9.
SELECT ∗
FROM SUPPLIER, *PARTSUPP*, *LINEITEM*, *ORDERS*, *NATION*
WHERE PS_SUPPKEY = L_SUPPKEY
AND L_ORDERKEY = O_ORDERKEY
AND S_SUPPKEY = L_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND S_NATIONKEY =3;
Actual result size: 10,111,600.
Cost of QEP determined by the R-ACM: 13718.
Final result size estimated by the R-ACM: 9,934,910.
Cost of QEP determined by the Equi-depth: 14856.
Final result size estimated by the Equi-depth: 9,624,692.

The corresponding screenshots for the two QEPS chosen for Q9 are given in Figure 8.

QEP of Query Q9 for Equi-depth



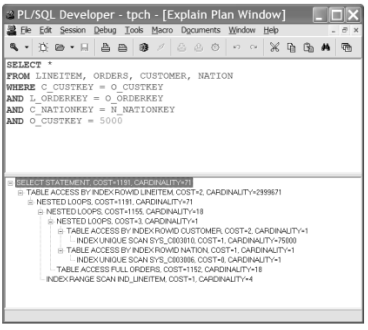
QEP of Query Q9 for R-ACM

Fig. 8. The QEPs Chosen for the Sample Query 9

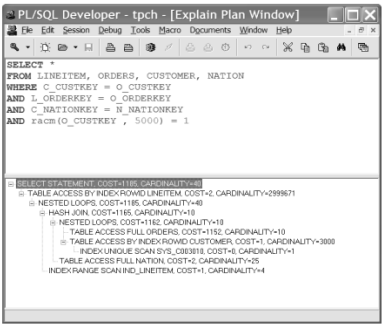
6. Example 5.2.g : SQL Query for Q10.
   SELECT ∗
   FROM LINEITEM, *ORDERS*, *CUSTOMER*, *NATION*
   WHERE C_CUSTKEY = *O_CUSTKEY*
   AND L_ORDERKEY = *O_ORDERKEY*
   AND C_NATIONKEY =*N_NATIONKEY*
   AND O_CUSTKEY = 5000;
   Actual result size: 39.
   Cost of QEP determined by the R-ACM: 1185.
   Final result size estimated by the R-ACM: 40.
   Cost of QEP determined by the Equi-depth: 1191.
   Final result size estimated by the Equi-depth: 71.

   The corresponding screenshots for the two QEPS chosen for Q10 are given in Figure 9.

7. Example 5.2.h : SQL Query for Q11.
   SELECT ∗
   FROM PARTSUPP, *SUPPLIER*, *NATION*
   WHERE PS_SUPPKEY = *S_SUPPKEY*
   AND S_NATIONKEY = *N_NATIONKEY*
   AND S_NATIONKEY = 3;
   Actual result size: 16,800.
   Cost of QEP determined by the R-ACM: 861.
   Final result size estimated by the R-ACM: 16,560.
   Cost of QEP determined by the Equi-depth: 908.
   Final result size estimated by the Equi-depth: 16,043.



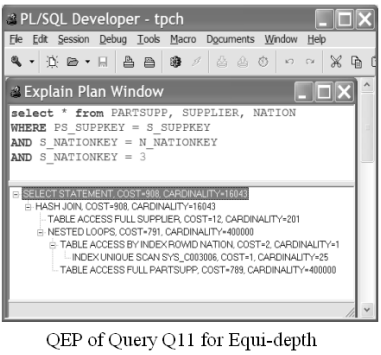
QEP of Query Q10 for Equi-depth



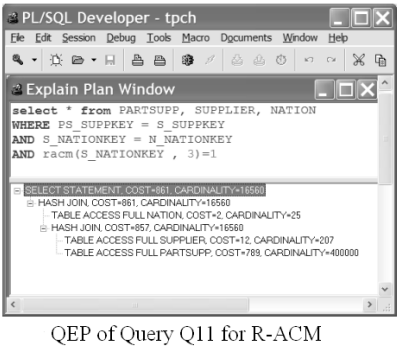
QEP of Query Q10 for R-ACM

Fig. 9. The QEPs Chosen for the Sample Query 10

The corresponding screenshots for the two QEPS chosen for Q11 are given in Figure 10.



QEP of Query Q11 for Equi-depth

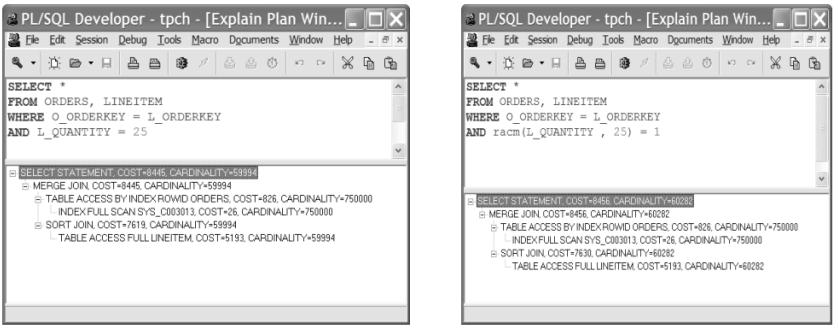


QEP of Query Q11 for R-ACM

Fig. 10. The QEPs Chosen for the Sample Query 11

8. Example 5.2.i : SQL Query for Q12.
   SELECT ∗
   FROMORDERS, *LINEITEM*
   WHERE O_ORDERKEY = *L_ORDERKEY*
   AND L_QUANTITY = 25;
   Actual result size: 60,251.
   Cost of QEP determined by the R-ACM: 8456.
   Final result size estimated by the R-ACM: 60,282.
   Cost of QEP determined by the Equi-depth: 8445.
   Final result size estimated by the Equi-depth: 59,994.

   The corresponding screenshots for the two QEPS chosen for Q12 are given in Figure 11.



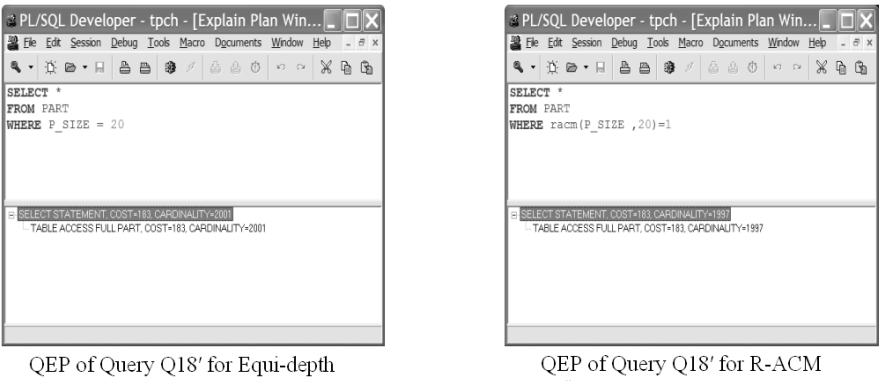
QEP of Query Q12 for Equi-depth

QEP of Query Q12 for R-ACM

Fig. 11. The QEPs Chosen for the Sample Query 11

9. Example 5.2.j : SQL Query for Q18'.
   SELECT
   FROM PART
   WHERE P_SIZE = 20;
   Actual result size: 1,997.
   Cost of QEP determined by the R-ACM: 183.
   Final result size estimated by the R-ACM: 1,997.
   Cost of QEP determined by the Equi-depth: 183.
   Final result size estimated by the Equi-depth: 2,001.

   The corresponding screenshots for the two QEPS chosen for Q18' are given in Figure 12.



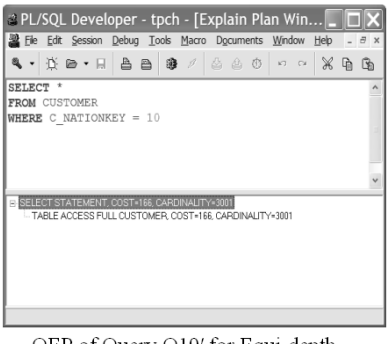
QEP of Query Q18′ for Equi-depth

QEP of Query Q18′ for R-ACM

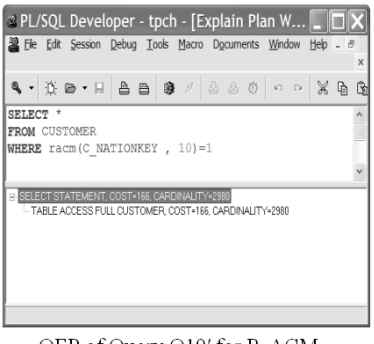
Fig. 12. The QEPs Chosen for the Sample Query 18'

10. Example 5.2.k : SQL Query for Q19'.
    SELECT
    FROM CUSTOMER
    WHERE C_NATIONKEY = 10;
    Actual result size: 2,961.
    Cost of QEP determined by the R-ACM: 166.
    Final result size estimated by the R-ACM: 2,980.
    Cost of QEP determined by the Equi-depth: 166.
    Final result size estimated by the Equi-depth: 3,002.

    The corresponding screenshots for the two QEPS chosen for Q19' are given in Figure 13.



QEP of Query Q19′ for Equi-depth



QEP of Query Q19′ for R-ACM

Fig. 13. The QEPs Chosen for the Sample Query 19'

Part I

Databases and Information Systems Integration

Improvement of Software Development Processes, Balancing Internal and External Organizational Aspects

Joseph Trienekens[1], Rob Kusters[1], Frans van Veen[2], Dirk Kriek[2], Daniel Maton[2], and Paul Siemons[3]

[1] Eindhoven University of Technology, Den Dolech 2, 5600 MB, Eindhoven, The Netherlands
[2] Quality House B.V., 3404 LL IJsselstein, The Netherlands
[3] Metrific, 3506 GT, Utrecht, The Netherlands

Abstract. Actual results of software process improvement projects happen to be quite disappointing in practice. Although many software development organisations have adopted improvement models such as CMMI, it appears to be difficult to improve software development processes in the right way. This paper presents a new approach to determine the direction of improvement for an organisation. This approach is based on literature research as well as an empirical investigation among software development organisations in The Netherlands. The results of the research show that software development organisations can be classified on the basis of their internal and external entropy, c.q. the level of (dis)order in the business system and its environment. Based on a possible imbalance between the internal and external entropy, directions for software process improvement can be determined. As such the new approach can complement and improve the application of current software process improvement methodologies, e.g. CMMI.

Keywords: Software process improvement, entropy, empirical research results.

1 Introduction

To be competitive in the current economy, more and more software development organisations strive at an improvement of their processes. This is based on convictions that an improvement of their software engineering processes ultimately will lead to an increased quality of the products. Over the last ten years several improvement models have been developed that act as frameworks and tools to improve software development processes [1]. Models such as the Capability Maturity Model (CMM, CMMI), [2], [3] are being applied increasingly in the software engineering domain. Starting point for process improvements is a so-called process assessment, i.e. an analysis and subsequent determination of the current performance level of the software development process. On the basis of the outcome of an assessment an improvement project can be initiated. However, the actual results of improvement projects are often quite disappointing in practice (Kusters and Trienekens, 2005). Case study reports and success factor investigations offer until now only a limited insight in the way software development organisations can and/or should try to improve their software development processes.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 75–85, 2008.

A closer look at assessment and improvement methodologies shows that they are strongly oriented on the internal processes of an organisation. They focus for example on internal organisational factors such as management commitment, on internal human factors such as training, on internal process factors such as project planning, on internal technical factors such as tool support, and on financial factors such as project budgets [2], [3]. As a consequence of this internal orientation it can be concluded that current software process improvement approaches do not take into account the influence of the environment of a software development organization on the way the software development processes could be improved [4].

This paper introduces a new approach for software process improvement. This approach uses as kernel concept the distinction between the internal and external entropy of a software development organisation. Entropy is a cybernetic concept that stands for the level of disorder of a system [5]. The main assumption in this paper is that an organisation should be aware of its own internal and external disorder. Differences between the internal and external disorder should play a role in the process of chosing a direction for process improvement. In this paper the theoretical concepts of internal and external entropy have been made operational. This has been done by an assessment of a selection of well-defined internal and external factors of a software development organisation. Based on an analysis of these internal and external factors the differences between the internal and external entropy, the so-called 'improvement space' of an organisation, is determined. Within its own 'improvement space' an organisation then has to strive at a balance between their internal and external entropy. This should be the main theme for their process improvement activities, and the usage of improvement models such as CMMI should be based on that theme.

This paper deals in chapter 2 with software process improvement, its limitations and the concept of entropy. Chapter 3 focuses on the metrification and quantification of entropy. In chapter 4 the results are presented of an empirical research project in that the entropy of eleven software development organisations has been measured. Finally, in chapter 5 conclusions and recommendations for future research are given.

2 Spi and the Necessity of Taking Both Internal and External Organisational Factors as Starting Point

In the software engineering domain, improvement activities are known as Software Process Improvement (SPI). SPI is being defined as:

The measurement-based improvement of the performance of the software development process aimed at delivery in due time, against agreed budgets, and with the required quality.

A well-known and broadly accepted model for the improvement of software development processes is the Capability Maturity Model (CMM). CMM prescribes a specific sequence of improvement activities that has to be followed to reach higher levels of software process maturity. CMM and comparable maturity models are used on the one hand to strive at a standardisation of processes and on the other hand to realise a decrease of product failures by eliminating their causes. However, it is questionable

whether these improvement models are applicable for each and every organisation. For instance, an organisation that has to deal with an increasing turbulent market, and that has improved its development processes by defining and standardising them in detail, will face difficulties in responding to the dynamic and changing requirements of that market. As a result there will be a 'mismatch' between the strengths of the software developing organisation and the needs of the market. Examples of other factors in the environment of an organisation that can influence its behaviour are changing governmental and legal factors. However, these types of external factors are until now not addressed by software process improvement methodologies. This paper proposes that both internal and external factors should be taken into account during a process assessment and they have to play a role in the determination of process improvement activities.

In order to make external and internal factors applicable to process assessment, the concept of external and internal entropy is introduced. Entropy is a concept with that the internal and external situation of a system can be expressed [5]. A low entropy means a high level of order, or structure and stability, in a system. A high entropy means a low, or even chaotic, level. Table 1 gives some examples of characteristics of organisations with high and low internal and external entropies. These rather abstract characteristics are used as a basis for making the concept of entropy more operational.

Based on this type of characteristics, organisations can be considered as dynamic entities that are continuously moving and changing, and that have to strive at a balance between their internal and external entropy. For instance, in case a market is very dynamic, and a company in that market is, as a consequence, faced with a high external entropy, this company then has to be able to respond quick and adequately to the market, and in other words should be flexible.

Table 1. Some examples of characteristics of organisations with high and low entropies

High external entropy	Low external entropy
Environment is unstable and unpredictable	Environment is stable and predictable
Standardisation is less important than flexibility	Flexibility is less important than standardisation
High internal entropy	Low Internal entropy
Flexibility is more important than standardisation	Flexibility is less important than standardisation
Organisational processes are often executed in a chaotic way	Organisational processes are executed in a formal and predictable way.

To increase its own flexibility, an organisation could be forced then to increase its internal entropy so that it becomes able to respond to the (dynamic) market. Such an organisation has to drop for example certain standardisation rules and has to increase the level of decisional freedom of employees in the development processes. The dropping of standardisation rules and the increasing of decisional freedom of employees should become main themes for software process improvement in this type of organisations.

3 Measuring Entropy

To be able to use the concept of internal and external entropy in a practical way it is necessary to make entropy measurable. In accordance with [5] entropy is based on the number of states that a system can have. In order to make entropy operational we link the state of a system to basic size variables. Of course this link has to be validated in practice and elaborated further, but in this paper organisations are considered to be business systems, consisting of interrelated components. The entropy of such a business system increases with:

- an increasing number of components;
- an increasing number of interactions between the components;
- an increasing number of changes in the environment.

From the above we derive that two concepts are playing a central role in the determination of entropy, respectively complexity and dynamics. These two concepts can be described as follows:

- complexity is a measure of the number of interacting components;
- dynamics is a measure of the amount of changes over time.

3.1 Entropy and Its Relations to Complexity and Dynamics

Entropy can be made measurable by using the concepts of complexity and dynamics. In principle, measuring is the linking of a value to a parameter, by positioning the parameter on a measurement scale. From the four types of measurement scales: ratio, interval, ordinal and nominal, only the ordinal scale applies for the measurement of the concepts of complexity and dynamics (and subsequently for the measurement of entropy). The ordinal scale offers the possibility to position a parameter on a measurement scale with values low, average, and high. In our research we have chosen to measure complexity and dynamics by using a four value measurement scale with the levels: low, average-low, average-high and high.

Table 2. Levels of Complexity

Complexity	Definition
Low	Almost all parameters have a low score
Average-Low	Most parameters have a low score
Average-High	Most parameters have a high score
High	Almost all parameters have a high score

Measuring complexity is executed via a number of business system characteristics, such as the number of employees per business function, the number of functions per department, the number of departments in the organisation, etc. Based on numbers for these parameters, organisations can be classified. For example the number of employees can be used to classify an organisation as small (1–9), average-small (10–99), average large (100–999) or large (1000+).

Similarly the dynamics of a business system can be measured. For example a classification can be derived on the basis of the number of process changes per time-frame. To determine the number of changes of a specific parameter, such as employees, information is needed on the input, the output, and the current number of the employees. Other parameters are for instance departments, products, customers, suppliers, competitors etc.

Consequently a total score for entropy can be calculated by combining the scores for dynamics and complexity, see table 3.

Table 3. Determination of entropy on the basis of measurements of complexity and dynamics

Entropy		Complexity			
		Low	Average-Low	Average-High	High
Dynamics	Low	Low	Low	Average-Low	Average-Low
	Average-Low	Low	Average-Low	Average-Low	Average-High
	Average-High	Average-Low	Average-High	Average-High	High
	High	Average-High	Average-High	High	High

3.2 Entropy and Its Relations with Business System Aspects

To determine the internal and external entropy of a software development organisation a questionnaire has been developed. This questionnaire is based on the following structure. First the concept of business system has been subdivided in the three aspects of Market, Organisation and Factory. Subsequently these aspects have been elaborated. Respectively:

Market has as sub aspects: products, design and architecture; Organisation has as sub-aspects: processes and structure; and Factory has as sub aspects: people, resources and technology. The objective is that for each of the three aspects (and sub aspects) the complexity and dynamics can be determined. Table 4 summarises the basic structure of the questionnaire. By using the questionnaire, information is gathered about the (sub)aspects and subsequently about the complexity and the dynamics.

Table 4. (Sub)aspect of Market, Organisation and Factory

		Sub aspect	Complexity	Dynamics
External entropy	Market	Product, Design, Architecture		
Internal entropy	Organisation	Process and Structure		
	Factory	People, Resources and Technology		

Table 5. Example of a question about the complexity of the Market

COMPLEXITY (of market on the basis of the aspects product, design and architecture)
How many parties are involved in the development of the software products (e.g. as supplier of sub-architectures, -designs and/or components)
Quantification: How large is the involvement of third parties? - percentage components delivered? - number of external parties that contributes to an architecture and/or design? - amount of time that external parties contribute?
What is the average effort that external parties contribute? - average number of employees per external party /number of employees intern - total number of employees of external parties /total number of employees in the project

On that basis the internal entropy (business system aspect Organisation and Factory) and the external entropy (business system aspect Market) can be determined. To illustrate the questionnaire and its questions, in table 5 an example is given of a question that addresses the complexity of the Market.

The questions about the business system aspect Market, lead to scores with that the external entropy can be determined. The questions about the business system aspects Organisation and Factory lead to scores on which basis the internal entropy can be determined. By comparing the scores of the external entropy and the internal entropy it becomes possible to position organisations in a two-dimensional matrix, see Figure 1.

A difference between the internal and external entropy shows, what we would like to call, the 'improvement space'. The concept of 'improvement space' indicates that it is possible to strive at a balance between internal and external entropy. The larger the distance an organisation has to the balance-line, the larger the space for improvement is, see Figure 1.

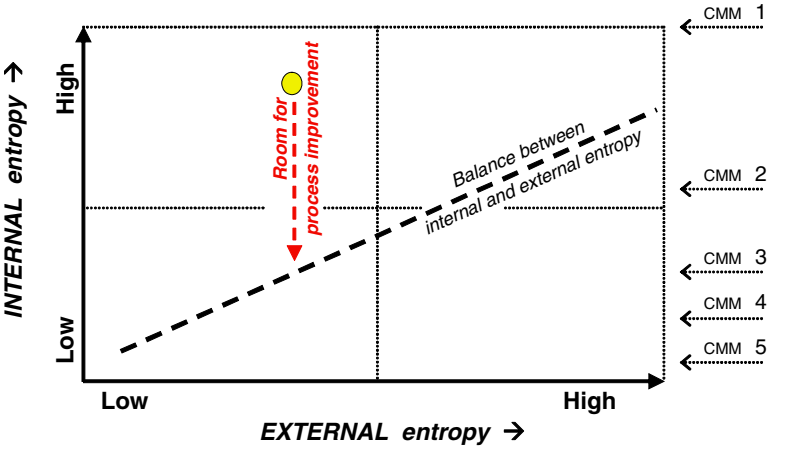


Fig. 1. Space for improvement for software development organisations

In Figure 1 the balance-line is represented by the dotted black line. The position of this balance-line is currently based on the usage of the data collected (i.e. the eleven companies). This means that we assume that on the average these companies have a 'certain' balance between internal and external entropy. It has to be emphasised that this is just a first step to determine the balance-line. Further research, and preferably more data, is needed to determine the balance-line more precisely.

The yellow bullet represents an organisation with a particular distance to the balance-line. This organisation has a relatively high level of internal entropy and a relatively low level of external entropy. Such an organisation should emphasise in its software process activities a decrease of its internal entropy, for example by striving at a certain level of standardisation of its processes. By doing so, the internal entropy will decrease until a balance is reached. Rather intuitively, on the vertical dimension to the right of Figure 1, the CMM-levels are given. Notice that on CMM5, which is the highest maturity level, the internal entropy is extremely low. This means that a software development organisation has an extremely high level of internal organisational order.

4 Results of the Research: Space for Improvement for Software Development Organisations

The questionnaire has been applied in an empirical research project in that eleven software development organisations in The Netherlands have been visited for the execution of interviews. In each organisation both a responsible person for software process improvement and a representative product manager have been interviewed. The software process improvement experts, with a strong internal orientation, had more than five years experience in the field. The product managers, with a strong orientation on the environment (or: external Market factors) of the organisation had about 10–15 years of relevant experience. The structured interviews took about 2 hours per interview. The results, answers to the questions, have been elaborated and have subsequently been reviewed by the interviewed persons.

The research has lead to a number of results. In the introduction of the interview sessions the interviewed persons have been asked to answer some more general questions. One of these questions was to describe the level of experience with software process improvement of the organisation. In section 4.1 we present first these experience levels. In section 4.2 we will present the entropy scores of the interviewed organisations.

4.1 Experience with Process Improvement

Answers to questions regarding the experience with software process improvement have lead to Figure 2. In total 17 persons could answer this question, see Figure 2. From this it appeared to be that about 80% of the visited organisations had experience with SPI.

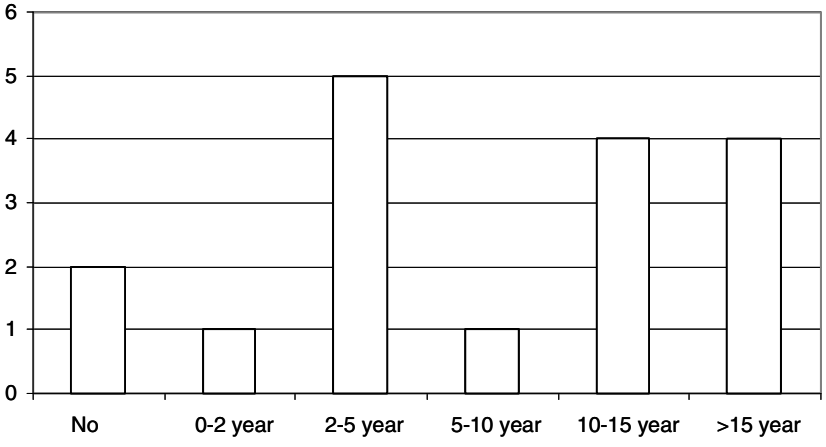


Fig. 2. Experiences with Software Process Improvement

4.2 Positioning Organisations on the Basis of Their Entropy Scores

Based on answers from the questions regarding the complexity and the dynamics of the three aspects Market, Organisation and Factory, the internal and external entropy could be determined. The following Figure 3 shows selected results.

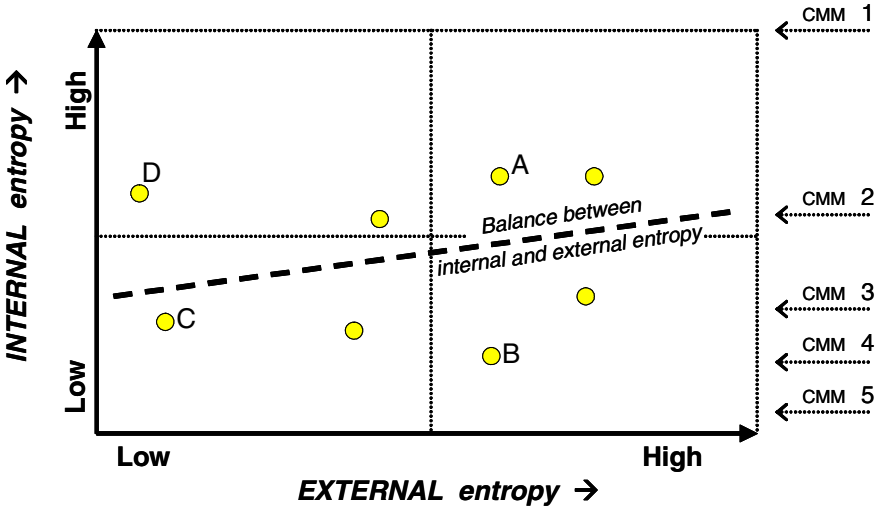


Fig. 3. Positioning of software development organisations on the asis of their internal and external entropy scores

Table 6. Characteristics of Organisation A

Organisation A	Complexity	Motivation	Dynamics	Motivation
Market	H	Various products for various markets	H	Product portfolio changes rapidly
Organisation	H	Many interrelations with suppliers	H	Many changes in collaborative processes
Factory	H	Large diversity in resources and people (skills)	H	Emerging new technologies

Table 7. Characteristics of Organisation B

Organisation B	Complexity	Motivation	Dynamics	Motivation
Market	H	Complex product portfolio	HH	High pressure of market which asks for lead time reduction in combination with innovative products
Organisation	L	Rather rigid formal matrix organisation	L	Stable structure and standardised processes
Factory	L	High level of standardisation	HH	Many job changes, fast emerging technologies.

Table 8. Characteristics of Organisation C

Organisation C	Complexity	Motivation	Dynamics	Motivation
Market	L	Stable and mature market with restricted product portfolio	L	Long lead time per product
Organisation	M	High level of maturity of business processes	LL	Rather stable processes, only few changes per time interval
Factory	H	High level of standardisation of all resources is a necessity	L	Few changes in resources to be applied

Based on the above mentioned scores for the business system characteristics of the software development organisations, their scores for the internal and external entropy could be determined.

These scores have then been used to position each software development organisation in the internal/external entropy balance Figure, see Figure 3.

From the four tables 6, 7, 8 and 9 it can be concluded that organisations can have totally different characteristics regarding their internal/external entropy balance. As a consequence different organisations should follow rather different directions regarding the improvement of their software development processes. For example, it is possible that an organisation wants to strive at a decrease of its internal entropy by

Table 9. Characteristics of Organisation D

Organisation D	Complexity	Motivation	Dynamics	Motivation
Market	L	Limited product portfolio, relatively low complexity of design and architecture	M	Influence of customers on product portfolio increases.
Organisation	L	Flat, formal organisation	HH	Many reorganizations necessary due to market pressure
Factory	H	Low level of standardisation	H	Allocation of (new types of) resources in the business processes is emerging

means of a standardisation of its development processes. By doing so, this organisation can reach a more balanced internal/external entropy situation. Of course it is also possible that an organisation wants to influence its external entropy by focusing on new emerging markets (i.e. increasing its external entropy) or by focusing on existing stable markets (i.e. decreasing its external entropy). The final objective of these 'improvement directions' or 'improvement themes' is a more balanced situation regarding the internal and external entropy of the software development organisation.

5 Conclusions

This paper presents the hypothesis that software development organisations can be classified and can be positioned on the basis of their internal and external entropy. Based on empirical research in eleven organisations the concept of entropy has proven to be fruitful for the characterisation of both the internal business system and its (external) environment. By using these internal and external characteristics, a possible imbalance can be determined for an organisation and a direction for improvement can be pointed out.

In this paper the characteristics of selected software development organisations have been determined on the basis of questionnaires that have been used in formal structured interviews. The concepts and terminology used has lead to positive reactions of the interviewed practitioners. The research results show that software development organisations can have quite different internal/external entropy characteristics, and this can lead to quite different software process improvement directions. Although the results gained are interesting we realise that further research is needed to validate the assumptions and constructs made. In particular we will investigate further the measurability of the internal and external entropy on the basis of the concepts of complexity and dynamics. Other main issues in our near-future research are the more precise determination of the entropy balance-line, and also the determination of the relation between CMM-levels and the internal/external entropy levels, which is currently still intuitively assumed. The final result at that we aim at, is a method (a.o. systematic analysis) and tool (i.e. automated questionnaire), to determine the improvement space of software development organisations that strive at efficient and effective software process improvement.

References

1. Balla, K., Bemelmans, T., Kusters, R.J., Trienekens, J.J.M.: Quality through Managed Improvement and Measurement (QMIM): towards a phased development and implementation of a Quality Management System for a Software Company. Software Quality Journal 9, 177–193 (2001)
2. Carnegie Mellon Software Engineering Institute, http://www.sei.cmu.edu/cmmi/
3. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability Maturity Model for Software, Version 1.1., SEI-CMU-93-TR-25 (1993)
4. Kusters, R.J., en Trienekens, J.J.M., Hassoldt, W.: On the business impact of software process improvement. In: Martin, D.C. (ed.) Proceedings of the IEEE 26th international computer software & applications conference, pp. 59–67. IEEE Computing Society, Los Alamitos (2002)
5. Boltzmann, L.: Entropie und Wahrscheinlichkeit, Ostwalds Klassiker, Bd 286 (2000)

Measuring Enterprise Resource Planning (ERP) Systems Success: A Structural Equation Modeling Approach

Princely Ifinedo

Department of Financial and Information Management, Shannon School of Business
Cape Breton University, Sydney, Nova Scotia, Canada
princely_ifinedo@cbu.ca

Abstract. Enterprise Resource Planning (ERP) systems are diffusing globally, and it is important to measure the success of such software in adopting firms. Evidence suggests that firms investing huge sums of money in information systems (IS) sometimes do not assess the success of such systems for a variety of reasons, including the lack of knowledge about what to assess. Also, the IS success evaluations research area is varied, often providing little succour to practitioners. ERP systems success assessment is just beginning to surface, and this paper discusses an effort towards extending an available success measurement model. Essentially, two relevant success dimensions not included in the model proposed by Gable and colleagues [9] and [30] were incorporated and tested using criterion analysis and structural equation modeling technique. The implications of the study's findings for practice and research are discussed.

Keywords: ERP systems success, IS success, Measurement, Evaluations, Structural equation modeling, Firms, Success dimensions.

1 Introduction

Enterprise Resource Planning (ERP) systems are complex business information technology (IT) packages designed to integrate business processes and functions by permitting the sharing of common data and practices in a real-time environment [5], [6] and [32]. Organizations adopt them for a variety of reasons, including the replacement of legacy systems and cost reductions [1], [5] and [6]. Assessing the success of ERP in organizations is difficult because of its complex nature [9], [28], [29], and [30].

In fact, some firms appear to have given up hope of evaluating the benefits or success of their ERP due to a lack of knowledge regarding such exercises [12]. In-depth interviews in 7 case companies regarding how they evaluate the success of their ERP software revealed that only 3 had any formal evaluations, the others indicated that they do not carry out such evaluations; yet, almost all these firms have adopted costly top brands ERP systems. The observations in this work are similar to those made by Kumar [18] and Seddon et al. [27] where these researches discussed the poor state of IS systems evaluations in organizations. Seddon et al. [27, p. 11) concluded, "…firms do not conduct rigorous evaluations of all their IT investments" perhaps due to a lack of knowledge in such areas. Participants in the study echoed a similar view (see [12]).

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 86–97, 2008.

The assessment of the value or success of IT systems has been a critical issue for both practitioners and researchers [15], [27], and [7]. Indeed, over the past three decades, evaluating the value and success of IT systems for organizations has been a recurring topical issue across both communities [7], [14], and [21]. In response to the topicality of the issue, various assessment approaches have surfaced. One stream of research focuses on the use of attitudinal and subjective measures [4] and [14], while another utilizes financial and objective parameters (see e.g. [7] and [10]). In both instances, understanding the success of the IT systems could be limited when the dimensions and measures of success are restrictive [9], [10], and [21]. Grover et al. [10] argue for measures that are more comprehensive to be used for information systems (IS) success studies. Perhaps it was the plethora of IS success assessment approaches that led Keen [15] to seek clarification of the "dependent variable." In response, DeLone and McLean (D&M) [7] developed an integrated, multi-dimensional, and inter-related IS success model that is now the dominant model for IS evaluation research [26]. Please see Figure 1 for the D&M model.

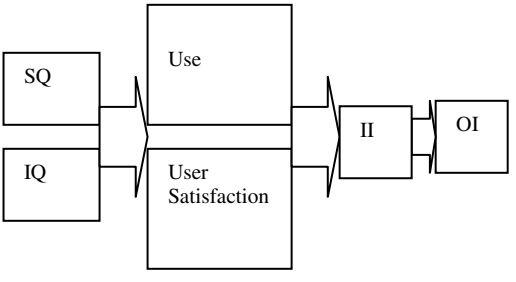


Fig. 1. DeLone & McLean [7] IS Success Model

Further, in developing their ERP success measurement model, Gable and colleagues [9], [28], [29], and [30] suggest that perhaps one of the reasons why there are mixed results reported with regard to IS success research is the utilization of limited or narrowly defined success dimensions. It comes as no surprise, therefore, that practitioners espouse a lack of knowledge regarding assessing the success of their acquired IT systems when the research community appears to lack a consensual approach on "what to assess?"

Thus, the lack of knowledge for some practitioners about what to measure or assess in the context of ERP systems [12], is the primary motivation for this study. As previously mentioned, Gable and colleagues [9], [28], and [30] have stepped up to this challenge, and this study only serves to complement their effort. In advancing the knowledge in this area, the following questions were specifically asked: Are the dimensions of success represented in the ERP success measurement model proposed by Gable and colleagues comprehensive? If otherwise, can the model be extended? The purpose of this study is to present an extended model that could be used by practitioners. The focus is on private organizations in contrast to the public sector organizations that Gable and colleagues studied. Ifinedo [12] cited cases where IT implementation and assessment issues for a government environment were noted to

be different from those in the private sector because of the profit oriented nature of the latter.

This research is conducted in Finland and Estonia - two small neighboring technologically advanced Northern European countries. Both countries share similar cultural values and orientations [13]. Finnish companies began adopting ERP in the late 1990s [8] and [12].

2 Background Information

Here, ERP systems success refers to the utilization of such systems to enhance organizational efficiency and effectiveness [1], [9], and [10], and it is different from ERP implementation success [12], [19], [20], [33], and [34]. The scan of the literature of ERP success research revealed that researchers either use narrowly defined measures [23], [33], [34], and [35] or elaborate on broad conceptualization of the concept (e.g., [33] and [19]. Some of the researchers used the end-user satisfaction instrument [4] and [14] that has been criticized for its limited scope [25]. Markus and Tanis [19] discussed ERP success by including performance metrics and outcomes, and noted that their "theoretical framework … is too broad in scope for direct empirical testing (Ibid, p. 200).

Gable and colleagues [9], [28], [29], and [30] provide perhaps the most comprehensive ERP systems success measurement model, to date, and others have used it (e.g., [31]). Gable and colleagues developed an additive model that redefines the dimensions in the original D&M IS success model. They noted that Seddon and Kiew (see [26]) tested paths in D&M model finding support for some and not for the others. In brief, Gable and colleagues eliminated (through multi-stage data collection and statistical analysis) the Use and User satisfaction dimensions. Arguments against dropping them are also available in the literature [25] and [26]. Furthermore, in their arguments for the mutual exclusivity of success dimensions, Gable et al. [9] suggested an overarching view of success in which "each measure [and/or dimension] only addresses one important aspect of IS success" (p. 578). In brief, the retained dimensions of ERP system success in Gable and colleague model are as follows: System Quality (SQ), Information Quality (IQ), Individual Impact (II) and Organizational Impact (OI). Please see Figure 2.

This study asks whether the model depicted in Figure 2 can be extended to include other relevant factors. To that end, the literature was consulted. Interviews in 7 case companies (private firms) in Finland and Estonia that had adopted ERP systems were conducted. In-depth discussions of this study are available elsewhere [12]. Evidence obtained from 16 senior personnel in these firms revealed that the cooperative role and quality of service of the ERP providers (vendors and consultants) is linked to the overall success of their ERP. One interviewee captured the views of others when he commented: "*As for me, I consider the support from the vendor, their expertise and commitment levels to be critical to our ERP success*" (Head of IT, Estonian manufacturing firm).

In this light, it is believed that a more comprehensive ERP success model should incorporate the Vendor/Consultant quality dimension. The quality of ERP providers throughout the life span of any ERP acquisition is imperative, and is recognized in the literature (Davenport, 1998; Markus and Tanis, 2000; Somers et al., 2000; Ko et al.,

2005). Markus and Tanis (2000) highlighted "dependence on vendors" as a key issue in ERP implementations that differentiates these systems from other IT implementations. Recently, Ko et al. [17] underscored the crucial role that vendors/consultants play during ERP implementations. Vendors and consultants are grouped together because they represent an external source of expertise to the firm in ERP implementations. Moreover, Sedera et al. [29] noted that items used to denote consultant and vendor qualities loaded together. It is argued that the underlying philosophy of ERP systems that facilitates the harmonization and integration of organizational functions and departments [5], [6], and [19] makes a case for the incorporation of a dimension relating to the issue of inter-departmental or cross-functional impacts.

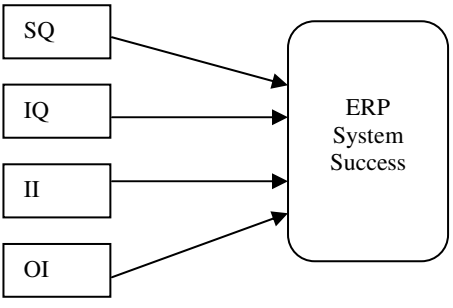


Fig. 2. Gable et al. [9] ERP Systesm Success Model

Along this similar line of reasoning, Myers et al. [22] and [23] argue that any IS success model should incorporate Workgroup Impact in light of the contributions made by work teams/groups toward organizational productivity, and these authors added it to the D&M model. Workgroup encompasses the sub-units and/or functional departments of an organization.

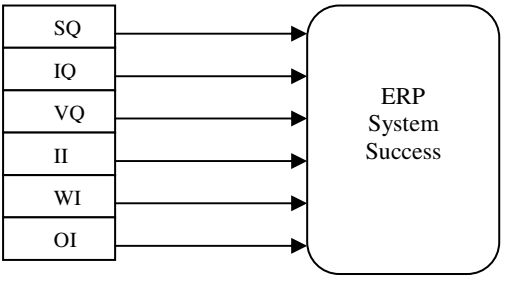


Fig. 3. The extended ERP systems success framework

Further, "interdepartmental co-operation" and "interdepartmental communication" ranked 3rd. and 6th. respectively in a study of 22 critical success factors (CSFs) of ERP implementation by Akkermans and van Helden [1]. Other CSFs studies have produced comparable results (see [8]). Thus, the conceptualization of ERP systems

success measurement model is shown in Figure 3 with two new dimensions: Vendor/Consultant Quality (VQ) and Workgroup Impact (WI).

3 Methodology

This study is a part of three-stage research effort using both qualitative and quantitative research approaches. Here, the report the main survey is reported. Admittedly, it was impossible for us to determine the number of firms adopting ERP in Finland and Estonia due to the unavailability of such a sampling frame. Rather, sampled firms generated from local contacts, ERP User Groups and vendors lists, as well as published lists of Top Enterprises for 2004 for both countries was used. Firms were chosen by the researcher's ability to obtain contact addresses for key organizational personnel. 50 firms in Finland and 120 firms in Estonia were identified. In order to ensure data validity and reliability, four knowledgeable individuals completed the questionnaire prior to mailing it, and their comments helped us improve the quality. Respondents in the survey indicated agreement with statements using a 7-point Likert-type scale, where 1 = strongly disagree and 7 = strongly agree (the questionnaire is omitted due to space restrictions).

Since the unit of analysis of this study was at the functional and organizational levels only key organizational informants including chief finance officers, unit managers, and IT managers received a packet consisting of a cover letter, questionnaire, and a self-addressed, stamped envelope. 40% of the mailings were matched pairs (two questionnaires in the packet), and the recipients were encouraged to give one of the questionnaires to an appropriate person within their organization. It was felt that multiple respondents from one organization would enhance the validity of the study, as common source variance would be reduced. The other 60% included only one questionnaire. Subjects were encouraged to present views representative of their organization.

3.1 Result

The overall response rate is 9.5% (44 firms) combined for the two countries, namely, 29 and 15 firms for Finland and Estonia, respectively. In total, 62 individual responses: 39 from Finland and 23 from Estonia were received. Of which, there were 26 (42%) top-level management and 36 (58%) mid-level managers. These groups of respondents are among the most knowledgeable informants regarding ERP success [9], [30], and [12]. There were 35 (56.5%) men and 27 (43.5%) women in the sample. On average, they had 9 years of work experience in their respective organizations. Of the respondents, 40% had college degrees, and 43 (69.3%) were aged between 31 and 50 years. Of the 62 respondents, 33.9% of them had SAP in their organizations, 14.5% had Movex, 9.6% had Scala, 8.1% had Hansa, and the remaining 33.9% had other mid-market ERP including Concorde and Scala. The majority of firms implemented their ERP between 1998 and 2002. Responses from a wide range of industries, including manufacturing, financial services, retail businesses were received. The sample included 15 small firms, 25 medium-sized firms, and 22 large

companies using the workforce categorization guidelines provided by the European Commission, see [12].

3.2 Instrument Development and Validity

The research instrument was developed from measures and constructs that have been validated in the literature [9], [28], [29], and [30]. Although for one construct – Workgroup Impact – the guidelines provided by Myers et al. [21] and [22] was used, and information garnered from the case interview [12. Forty-five (45) measures for the 6 dimensions and 3 measures were used to assess the ERP systems success construct. SQ comprised 10 measures such as "Our ERP has accurate data", and IQ comprised 9 measures, including "The information on our ERP is understandable" [6] and [9]. VQ consists of 5 measures, including "Our ERP vendor/consultant is credible and trustworthy" [17]. II consists of 6 measures, including "Our ERP improves individual productivity" [7], [9], and [22]. WI comprised 7 measures, including "Our ERP helps to improve workers' participation in the organization" [12], [21], and [22]. "Our ERP reduces organizational costs" is among the 8 measures included in the OI dimension. The ERP systems success construct has 3 measures from Gable et al. [9] (see below: Criterion analysis). The content validity of the study is enhanced over stages in the study, including the pilot test. Regarding the reliability of the measures, the Cronbach Alpha for each dimension ranged from 0.769 to 0.942, which is above the 0.70 limit recommended by Nunnally [24], thus indicating a reasonably high reliability of the research measures.

4 Data Analysis

4.1 Additivity of the ERP Systems Success Dimensions

Following guidelines in Gable et al. [9], the additive nature of the extended model was assessed by investigating the criterion validity of the measures in the research instrument. ERP systems success was assessed with the following three statements: (A) "Overall, the impact of our ERP on me has been positive," (B) "Overall, the impact of our ERP on my workgroup has been positive," and (C) "Overall, the impact of our ERP on my organization has been positive." To assess the content and the criterion validity of ERP success, we computed the following composite measures: (D) "criterion average" is the average of the three criterion items, and (E) "dimensions average" is the average of the six success dimensions. Table 1 shows the correlation of (A), (B), (C), and (D) with the six dimensions and their average (E). Gable et al. [9, p. 585] stated, "The extent to which each dimension or the dimension average correlates with the criterion scores is evidence of their criterion validity" (see also, Kerlinger [16]).

The correlations are significant at the 0.01 level (two-tailed), with the exception of the correlation between "Organization Impact" and "Impact on Individual," which is 0.70. The three largest correlations are for (A), (C), and (D) with (E), which are respectively 0.70, 0.72, and 0.74. Consistent with Gable et al. (2003), the largest correlation (0.74) is between (D) criterion average and (E) dimension average, which suggests that (D) and (E) are the strongest measures of overall ERP success. Gable et al.

[9, p. 585] noted, "that the dimension average yields the largest correlation with all the criteria further supports the view that the dimensions are additive, and thus when combined yield a stronger overall measure of success than possible from any single dimension." In this regard, the data supports the work of Gable et al. [9].

Table 1. Correlations: Criteria and Dimensions

	Dimension	**A**	**B**	**C**	**D**
1	SQ	.55	.54	.64	.61
2	IQ	.59	.58	.63	.64
2	VQ	.41	.42	.40	.43
4	II	.51	.57	.60	.59
5	WI	.60	.58	.57	.62
6	OI	.70	.61	.67	.69
E	Dimension Average	.70	.68	.72	.74

Legend: A: Impact on Individual, B: Impact on Workgroup, C: Impact on Organization, D: Criterion Average.

PLS Graph 3.0 was used to assess the extended ERP success measurement model. The PLS (Partial Least Squares) procedure is a second-generation multivariate technique used to estimate structural models [2] and [3]. This approach is suitable for this study for its small-sized nature, and the developing knowledge regarding the additive nature of IS success measurement. PLS is capable of testing complex models consisting of multiple interactions measured with multiple indicators. PLS recognizes two components of a casual model: the measurement model and the structural model [2] and [3].

The measurement model consists of relationships among the conceptual factors of interest (the observed items or variables) and the measures underlying each construct. This model demonstrates the construct validity of the research instrument, i.e. how well the instrument measures what it purports to measure. The two main dimensions are the convergent validity (composite reliability) and the discriminant validity. PLS Graph 3.0 computed the composite reliability of each dimension or construct. The composite reliability of each construct in the model with the highest predictive power in this study are as follows: SQ - 0.73; IQ - .62, VQ - 0.51, II - 0.58, WI - 0.50, OI - 0.63, and ERP success - 0.77 (please see the discussions below). This is adequate for this study [11]. The discriminant validity is assessed by checking the extent to which items measure a construct. This is assessed by checking the square root of the average variance extracted (AVE) for each construct. In no case was any correlation between the constructs equal to or greater than the squared root of AVE [2]. This suggests that the used measures are distinct and unidimensional (The result is omitted due to space restrictions, but available upon request). Thus, it can be said that the convergent and discriminant validity of the data are psychometrically sound and adequate for an explanatory study such as this one [2] and [11].

The structural model gives information as to how well the theoretical model predicts the hypothesized paths or relationships. PLS Graph 3.0 provides the squared

multiple correlations (R^2) for each endogenous construct in the model and the path coefficients. The R^2 indicates the percentage of a construct's variance in the model, while the path coefficients indicate the strengths of relationships between constructs [2] and [3]. PLS does not generate a single goodness of fit metric for the entire model, unlike other structural modeling software, but the path coefficients and the R^2 are sufficient for analysis [2] and [3].

4.2 Alternative Models

The examination of alternative models in structural modeling could facilitate insights [4], [11], and [30]. Thus, alternative models - Model 1 to Model 6 – were developed, and their path coefficients and R^2 s checked. The details of these models are shown in Table 2 (See also the Appendix for their illustrations). The R^2 of Models 1, 2, 3, 4, 5, and 6 respectively are 0.335, 0.366, 0.050, 0.305, 0.362, and 0.316. Clearly, Model 2 has the best R^2 suggesting its relative strength in predicting ERP success in comparison to the other models. It has to be noted that all the models but Model 3 explained more than 30% of the variance in the ERP success model, which is adequate for this study. The extended ERP success framework represented in Model 2 suggests that ERP success is a second-order factor. This is consistent with results in the work of Gable and colleagues [9] and [30]. Even though the extended ERP success model herein has more dimensions than do Gable et al. [9], the conclusions seem to be comparable.

Further, Figure 4 shows the path coefficients in Model 2. Chin [2] recommends that path coefficients should be at least 0.20, and ideally above 0.30 to be considered meaningful. Apparently, SQ and OI predict "success" more than do any other dimensions with their relatively better path coefficients. Again, this result corroborates the results by [28] in which these two dimensions were noted as the most important in assessing ERP success. These researchers sampled the views of key organizational stakeholders in 23 Australian public sector organizations using the four dimensions in the Gable et al. model [9].

Table 2. Correlations: Criteria and Dimensions

R^2	Structural Model	Description
$R^2 = 0.335$	Model 1	One first-order factor, with all the 45 items
$R^2 = 0.366$	Model 2	Six first-order factor (SQ, IQ, VQ, II, WI, OI), One 2nd order factor
$R^2 = 0.050$	Model 3	Six first-order factor, Two 2nd order factors, One 3rd order factor
$R^2 = 0.305$	Model 4	Four first-order factor (SQ, IQ, II, OI), One 2nd order factor Gable et al. [9]
$R^2 = 0.362$	Model 5	Five first-order factor (SQ, IQ, II, WI, OI), One 2nd order factor (without VQ)
$R^2 = 0.316$	Model 6	Five first-order factor (SQ, IQ, VQ, II, OI), One 2nd order factor (WI)

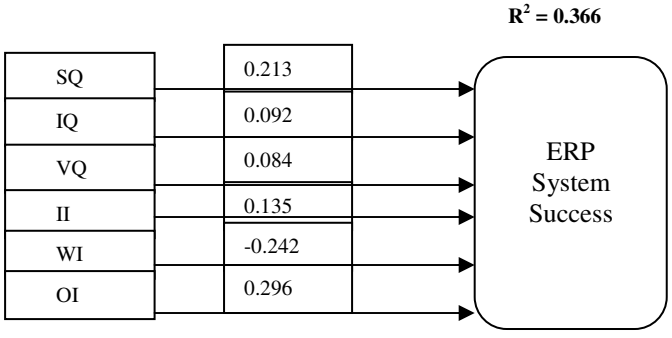


Fig. 4. Results of PLS graph 3.0 for model 2

5 Discussions and Conclusion

This paper discusses ERP success measurement model as proposed by Gable and colleagues [9], [28], and [30]. Specifically, the research study aimed at investigating whether the Gable and colleagues' model is comprehensive. It is found through literature review and interviews with case companies that their ERP systems success measurement model might be limited in scope as two important dimensions are not considered. To that end, this paper presents perhaps the first attempt at validating and extending their model, and in a different setting (private sector) and geographical location. Importantly, this paper draws from the issues of additivity and mutually exclusivity of ERP success measures discussed by Gable and colleagues as two relevant dimensions, i.e., Workgroup Impact and Vendor/Consultant Quality found to be relevant in the discourse, were incorporated.

With regard to research, this endeavor could entice further studies. The operationalized set of ERP dimensions (and measures) offers perhaps a more comprehensive model in the literature. The proposed ERP systems success measurement model (Figure 3) has sound psychometric properties as assessed through structural equation modeling technique, and criterion validity. Particularly, this effort might engender the development of an appropriate scale to assess ERP system success for adopting organizations. Further, support for the claim that ERP systems success is a second-order factor [30] was found. The data shows that a six-factor construct outperforms the one with four as proposed by Gable and colleagues.

Additionally, the extended framework offers other useful insights, for example, System Quality and Organizational Impact were found to be perhaps the two most important dimensions to watch out for in evaluating ERP systems success, this finding adds credence to a previous study [28] carried out in public sector organizations. Admittedly, the findings herein are not conclusive and further testing and refinements are expected. Future research might need to focus on utilizing confirmatory factor analysis as knowledge is accumulated in this area of research.

The study has implications for practice as well. As noted, this study is motivated by the need to present practitioners with guidelines for assessing the success of their ERP software. It is not claimed that the guideline is the final word regarding ERP success measurement, evaluation or assessment for ERP adopting firms; however, the

comprehensive list of success dimensions could be valuable especially for firms with no formal means of conducting such an exercise. It is worth noting that anecdotal evidence exists indicating that the research instrument is already in used for such purposes in the study's settings. Management can use the dimensions of Systems Quality and Organizational Impact of acquired systems in assessing the effectiveness or success of such technologies in instances where a more comprehensive instrument or formal evaluation techniques are not readily available. The proposed extended ERP success measurement framework or model could be modified for other enterprise systems, including Customer Relationship Management (CRM), and Supply Chain Management (SCM).

To conclude, the limitations of this study are highlighted as follows. The study is exploratory in nature, and the sample is not random. Personal bias cannot be ruled out, even when the respondents claimed to have presented an aggregate view on the selected issues for their respective organizations. The sample comprised of mixed ERP software, including top-brand names (e.g. SAP and Oracle) and mid-market products (e.g. Scala and Nova). It is possible that the heterogeneous nature of the ERP systems used for the study are limiting. Finally, the used sample consists of small, medium, and large companies. The diversity in the sample is good, but it may affect the study's findings. A homogenous sample of only large or small firms might yield results different from the ones discussed herein. Future studies could improve the findings of this study by addressing some of these limitations.

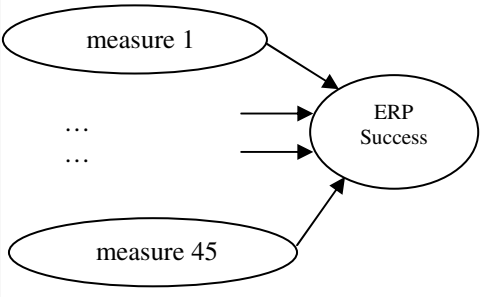
References

1. Akkermans, H., van Helden, K.: Vicious and virtuous cycles in ERP implementation: a case study of interrelations between critical success factors. EJIS 11, 35–46 (2002)
2. Chin, W.: Issues and opinion on Structural Equation Modeling. MIS Quarterly 22(1), vii–xvi (1998)
3. Chin, W.: Tutorial - Partial least squares for researchers: An overview and presentation of recent advances using the PLS approach. In: ICIS, Brisbane, Australia (2000)
4. Doll, W.J., Torkzadeh, G.: The measure of end user computing satisfaction. MIS Quarterly 12(2), 259–274 (1988)
5. Davenport, T.: Putting the enterprise into the Enterprise System. HBR 76(4), 121–131 (1998)
6. Davenport, T.: Mission Critical. Harvard Business School Press, Boston, MA (2000)
7. DeLone, W.H., McLean, E.R.: Information systems success: the quest for the dependable variable. ISR 3(1), 60–95 (1992)
8. Esteves, J., Pastor, J.: Enterprise resource planning systems research: An annotated bibliography. CAIS 7(8), 1–52 (2001)
9. Gable, G., Sedera, D., Chan, T.: Enterprise systems success: A measurement model. In: The 24th. ICIS, pp. 576–591 (2003)
10. Grover, V., Jeong, S.R., Segars, A.H.: Information systems effectiveness: The construct space and patterns of application. I&M 31, 177–191 (1996)
11. Hair Jr., J.F., Anderson, R.E., Thatham, R.L., Black, W.C.: Multivariate Data Analysis. Prentice-Hall International, Inc., Upper Saddle River, NJ (1998)
12. Ifinedo, P.: Do Organisational-technological contingency factors influence the perception of ERP systems success? An exploratory study in the Baltic- Nordic region of Europe. In: Proceedings of 4th. IBIMA, Lisbon, Portugal (2005)

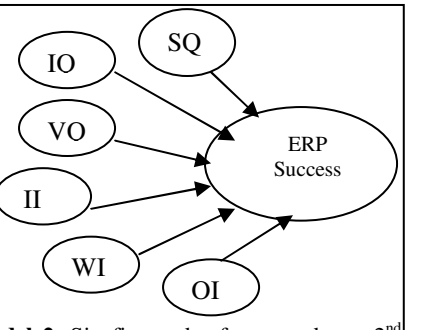
13. Ifinedo, P., Davidrajuh, R.: Digital divide in Europe: assessing and comparing the e-readiness of a developed and an emerging economy in the Nordic region. Electronic Government: An International Journal 2(2), 111–133 (2005)
14. Ives, B., Olson, M.H., Baroudi, J.J.: The measure of user information satisfaction. CACM 26(10), 785–793 (1983)
15. Keen, P.: MIS Research: Reference disciplines and a cumulative tradition. In: IICIS (1980)
16. Kerlinger, F.N.: Foundation of Behavioral Research. Holt Rinehart and Winston (1988)
17. Ko, D., Kirsch, J.L., King, W.R.: Antecedents of knowledge transfer from consultants to clients in enterprise system implementations. MIS Quarterly 29(1), 59–85 (2005)
18. Kumar, K.: Post implementation evaluation of computer-based information systems: current practices. CACM 33(2), 203–212 (1990)
19. Markus, L., Tanis, C.: The enterprise systems experience–from adoption to success. In: Zmud, R.W. (ed.) Framing the Domains of IT Research: Glimpsing the Future Through the Past, Pinnaflex Educational Resources, Inc., Cincinnati, OH (2000)
20. Markus, M.L., Tanis, C., Fenema, P.C.: Multisite ERP implementation. CACM 43(4), 42–46 (2000)
21. Myers, B.L., Kappelman, L.A., Prybutok, V.R.: A case for including work group productivity measures in a comprehensive IS assessment model. In: The 27th. Annual Meeting of the DSI (1996)
22. Myers, B.L., Kappelman, L.A., Prybutok, V.R.: A comprehensive model for assessing the quality and productivity of the information systems function: Toward a theory for information systems assessment. IRMJ 10(1), 6–25 (1997)
23. Nelson, K.G., Somers, T.M.: Exploring ERP success from an end-user perspective. In: The 7th. AMCIS (2001)
24. Nunnally, J.C.: Psychometric Theory, 2nd edn. McGraw-Hill, New York (1978)
25. Saarinen, T.: An expanded instrument for evaluating information system success. I&M 31, 103–118 (1996)
26. Seddon, P.B.: A re-specification and extension of the DeLone and McLean model of IS success. ISR 18(3), 240–253 (1997)
27. Seddon, P.B., Graeser, V., Willcocks, L.P.: Measuring organizational IS effectiveness: An overview and update of senior management perspectives. The DATA BASE for advances in IS 33(2), 11–28 (2002)
28. Sedera, D., Gable, G.G., Palmer, A.: Enterprise resources planning systems impacts: a delphi study of Australian public sector organisations. In: The 6th. PACIS (2002)
29. Sedera, D., Gable, G., Chan, T.: Measuring enterprise systems success: A preliminary model. In: The 9th. AMCIS, Tampa, Florida, USA (2003a)
30. Sedera, D., Gable, G.: A factor and structure equation analysis of the enterprise systems success measurement model. In: The 25th. ICIS (2004)
31. Sehgal, R., Stewart, G.: Exploring the relationship between user empowerment and enterprise system success measures. In: The AMCIS (2004)
32. Somers, T.M., Nelson, K.G., Ragowsky, A.: Enterprise resource planning (ERP) for the next millennium: Development of an integrative framework and implications for researcher. In: The 6th. AMCIS (2000)
33. Tan, C.W., Pan, S.L.: ERP success: the search for a comprehensive framework. In: The 8th. AMCIS (2002)
34. Wu, J., Wang, Y.: Measuring ERP success: the Key-users' viewpoint of ERP to produce a viable IS in the organization, Computers in Human Behavior (2005)
35. Zviran, M., Pliskin, N., Levin, R.: Measuring user satisfaction and perceived usefulness in the ERP context. JCIS 45(3), 43–52 (2005)

Appendix

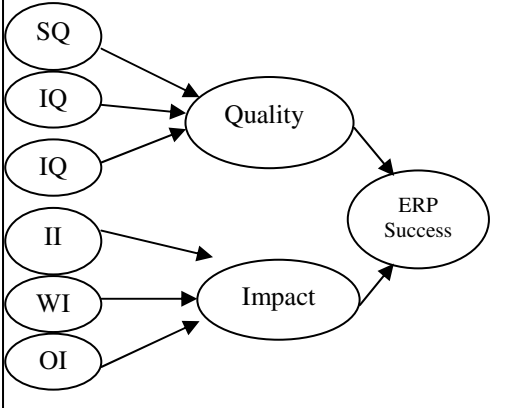
Illustrations of the alternative ERP systems success models.



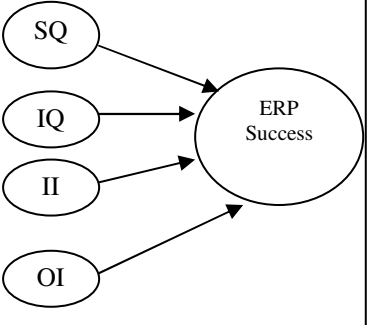
Model 1. One first-order factor, with all the 45 items.



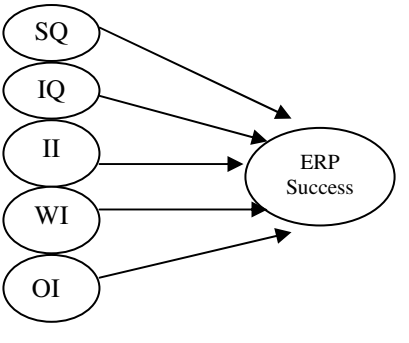
Model 2. Six first-order factor and one 2nd order factor.



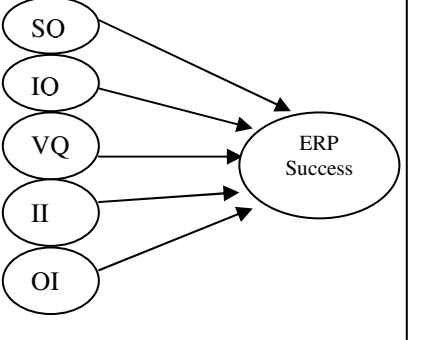
Model 3. Six first-order factor, Two 2nd order factors, one 3rd order factor.



Model 4. Four first-order factor (Gable et al.).



Model 5. Five first-order factor, one 2nd order factor (without VQ).



Model 6. Five first-order factor, one 2nd order factor (without WI).

Toward Data Compliance in Vaccine Industry: Interoperability to Align Business and Information Systems

Néjib Moalla[1,2], Abdelaziz Bouras[1], Gilles Neubert[1], and Yacine Ouzrout[1]

[1] CERRAL/LIESP, IUT Lumière Lyon 2. 160 Bd de l'Université, 69500, Bron, France
{Nejib.Moalla,Abdelaziz.Bouras,Gilles.Neubert,
Yacine.Ouzrout}@univ-lyon2.fr
[2] Sanofi Pasteur France. 1541, Avenue Marcel Mérieux, 69280, Marcy-L'étoile, France
Nejib.Moalla@SanofiPasteur.com

Abstract. The ultimate goal in the vaccine industry is product quality. However this quality can be altered by the use of a number of heterogeneous information systems with different business structures and concepts along the lifecycle of the product. Interoperability concepts may help to guarantee a certain correspondence and compliance between different product data. In this paper we focus on a particular compliance problem, between production technical data, represented in an ERP, and the corresponding regulatory directives and specifications, represented by the Marketing Authorizations (MA). The MA details the process for vaccine manufacturing according to the requirements imposed by health organizations such as Food and Drug Administration (FDA) and Committee for Medicinal Products for Human use (CHMP).

The proposed approach uses an interoperability framework which is based on a multi-layer separation between the organizational aspects, business contexts, and information technologies for each involved entity into the communication between the used systems.

Keywords: Vaccine industry, health Information Systems, ERP, Marketing Authorizations, Compliance, Interoperability.

1 Introduction

The vaccine industry is distinguished among process industries by the need to comply with regulatory constraints imposed by organizations such as WHO (http://www.who.int/) and FDA (http://www.fda.gov/) in the USA or AFSSAPS in France, etc**.** Further constraints are imposed by the conventions signed with national and international authorities, called Marketing Authorizations (MA) – Authorizations to Make to Market (AMM) in Europe – for the manufacture of vaccines.

In this operating context, the issue of product quality is one of high priority for a company in order to maintain its credibility compared to its customers. One of the key factors of quality is the good management of product data. In fact, product data come in several types and formats according to diverse business contexts and are supported by several heterogeneous information systems. The challenge is to enable communication

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 98–111, 2008.

among these systems and the process of guaranteeing the validity and the conformity of exchanged information. This challenge is seldom addressed systematically. Indeed, considering the complexity of information systems architectures in both regulatory and production contexts, there is a general tendency to check conformance only between the MA files and the Standard Working Instructions (SWI).

Our Scope in this paper covers the problem of communicating product data between information systems supporting the Marketing Authorizations (MA) and the ERP for structuring production data. Delivering a product according to its description in the MA requires the right information in the ERP. Otherwise, we risk manufacturing a non compliant product, to not deliver our product in time to respect customer commitments, and in final destroy these products and lose money.

We present in the second section a general description of vaccine industry and the evolution of vaccine product inside. The third section deals with the integration of interoperability concepts in vaccine industry and the fourth section presents a deployment of interoperability concepts to ensure data compliance in production information system. In the fifth section, we present a case study developed with Sanofi Pasteur company.

2 The Vaccine Industry

2.1 The Vaccine Supply Chain

Throughout its lifecycle, the vaccine product passes by following supply chain divisions:

Research division: Looks for new drugs or substances that can contribute to the creation of new drugs. At this stage conducted studies are reported and indexed in the form of technical reports.

Research & Development division: Conducts specific research, and is interested in the development of mixture processes of substances, tests and stability conditions of the final solution that can be defined as a drug. The information system is used to structure data about clinical trials and tests for validity. At this stage starts the definition of an explicit product structure.

Industrialization division: Defines the industrial infrastructure which will support the production of a defined product quantity on the basis of a definition of product solution. At this stage, we define technical data describing the product manufacturing operations and the used process and tools.

Production division: Deals with planning, scheduling and follow-up of production based on the data describing industrial infrastructure and product composition. At this stage, we identify static data compared to external dynamic data like work orders or those generated by the ERP such as buying orders of raw material.

Distribution division: Defines the conditions for handling the product for customer delivery in accordance with the description of the conditions of manufacture, which is given by R&D division. At this stage, product handling information is documented.

The specificity of the supply chain of vaccine industry dominates in the particular definition of its product, the vaccine [1, 2]. At the production stage, we can only control the manufacturing process and not the product itself. Indeed, the biological aspect of the active substances in the vaccines differs from products in the chemical industry (like pharmaceutical industry) by a very complex structure. The biological production consists of a mixture of molecular substances not always well identified. This specificity makes the control procedures, already different from one product to another, more delicate than complex.

2.2 The Vaccine Product

According to vaccine supply chain presentation, we can divide vaccine lifecycle in two huge stages: the vaccine design and the industrial production [3].

2.2.1 Vaccine Design

The development of a new vaccine presents a long term action that rapidly exceeds 10 years (Fig. 1). Starting from exploring phase, researchers look for new substances being able to contribute to the creation of a vaccine against a given disease. If a vaccine candidate emerges from the discovery research, preclinical tests allow its characterization for a better control of its behavior to generate the appropriate antigens.

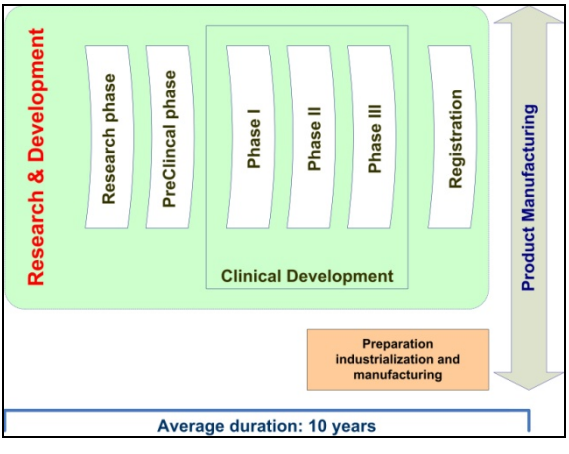


Fig. 1. Vaccine design

In clinical tests stage, several phases allow to better characterize the product by testing an increasing number of patients to determine the effectiveness, the safety and the harmlessness of the vaccine with the suitable intervals of injections. All information concerning clinical reports, tests, and control results are capitalized throughout these phases. To prove the reproductiveness and industrial capability, three consistency batches are manufactured.

Finally, to produce and market this new vaccine, it is necessary to prove to the health authorities (WHO http://www.who.int/, FDA in the USA http://www.fda.gov/,

etc) the utility of the vaccine [4, 5]. This passes by the submission of Marketing Authorization (MA) request to the health authority of destination country. The MA contains all information collected during the process of research and development. Once the product is approved, it can be manufactured and distributed inside this country.

2.2.2 Vaccine Production

The vaccine production is a rather complex process during from 6 months to 2 years. The production process, done by fixed batch size, passes by several states. For each state, we identify 3 tasks, the manufacturing, the quality control and the batch release. We can divide all manufacturing states into two great steps: biological manufacturing and pharmaceutical manufacturing (Fig. 2).

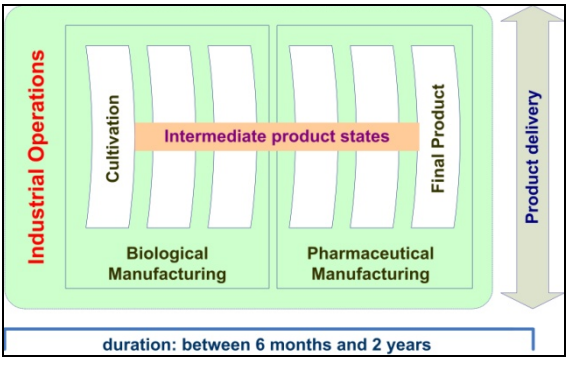


Fig. 2. Vaccine manufacturing

Biological manufacturing: covers from the vaccine valence manufacturing state until the final active substance state (monovalent) using production specificities retained and presented in the MA.

Pharmaceutical manufacture: consists in the mixture of active substances, their distribution in the appropriate dosage forms (syringe, blister, etc.) until the packaging. These operations must respect the sterility conditions, quality of air, quality of the final product, etc; all of them are specified in the MA.

Considering the complexity of the biological definition of the vaccine and the large quantity of data to define it, the management of the product data presents a big challenge in this context. In the vaccine design stage, product data are generally manually written data, so it's very complex to analyze these data and use them in other information systems like the ERP.

2.3 Vaccine Data Specifications

2.3.1 The Complexity of Medical Data Exchange

The pivotal problem of medical data is the absence of machine readable structures [6]. Most healthcare data is narrative text and often not accessible. Generally, related works [6] have a certain tendency to treat this problem in structuring drug and other

information using XML standards. This is generally made using topic Maps [7], but presenting a product XML data models and connecting them is not sufficient [8]. Some Standard for the Exchange of Product Model data (like STEP-ISO 10303) addresses this through formats and programming interfaces derived directly from domain-related information models written in the EXPRESS information modeling language. However, these formats and programming interfaces are predetermined [9], and not always well suited to current information processing technologies. We can find also Product Data Markup Language (PDML) [10] as an Extensible Markup Language (XML) vocabulary designed to support the interchange of product information among commercial systems (such as PDM systems) or government systems (such as JEDMICS), where the vocabularies are related via mapping specifications.

2.3.2 The Need of Production Data Reliability

Performing data mapping between regulatory and industrial product definition present a hard task that requires regrouping efforts from different sectors like regulatory affairs, industrial operations, information systems, etc.

The majority of vaccine industries are specialized in biologic development of vaccines. The implication of a deviation in manufacturing or subcontracting can run the gamut from very minor to catastrophic. Our challenge consists in delivering the right product data value through manufacturing states in the production information systems.

During manufacturing process, the product passes from one state to another. Each state may concern one or several components and we have to validate their corresponding specifications based on data coming from MA information system. The following Figure (Fig. 3) presents a hierarchical structure for a product in the ERP (Enterprise Resource Planning) system of the company.

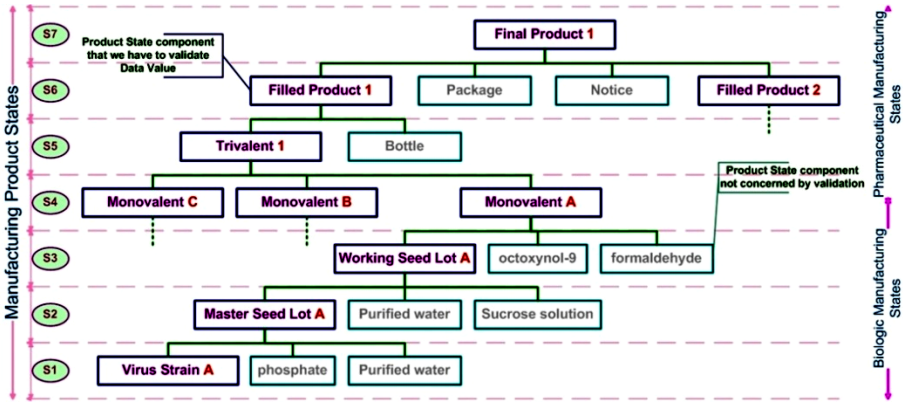


Fig. 3. Manufacturing product states and state components

When we have to validate in the ERP, it is necessary to find and validate product data value for each component through different product states.

3 Interoperability in Vaccine Industry

3.1 The Interoperability Framework

The IEEE standard computer dictionary defines interoperability as "the ability of two or more systems or components to exchange information and to use information that has been exchanged". The EU Software Copyright Directive [11] gives a similar definition and considers the interoperability between computing components as "the ability to exchange information and mutually to use the information which has been exchanged". This does not mean that each component must perform in the same way, or contain the same functionality as the other components– interoperability is not a synonym for cloning. Rather, interoperability means that components with different functionalities can share information and use it according to their needs.

The European Interoperability Framework definition identifies three separate aspects:

Organisational – is concerned with defining business goals, modelling business processes and bringing about the collaboration of administrations that wish to exchange information, but that may have a different internal organisation and structure for their operations.

Semantic – is concerned with ensuring that the precise meaning of exchanged information is understandable by any other application not initially developed for this purpose. Semantic interoperability enables systems to combine received information with other information resources and to process it in a meaningful manner.

Technical – covers the technical issues of linking up computer systems and services. This includes key aspects such as open interfaces, interconnection services, data integration and middleware, data presentation and exchange, accessibility and security services.

Identification and structuring of these interoperability types help to perform a better exchange between systems Therefore, it is necessary to identify the area of our investigation and its specifications: structures, business constraints, etc.

To achieve interoperability among divisions systems in collaborative enterprise, we consider three challenges [12]:

Heterogeneity, incoherent information, different systems and software infrastructures, different working practices, etc.

Flexibility, information reuse, following of variations in documents versions, etc.

Complexity, definition granularities, dependency between different components, etc.

Heterogeneity, flexibility and complexity must be managed at different levels:

Knowledge, approaches, methods and skills needed for innovation, shared languages.

Process, planning's, coordination and management of cooperative and interdependent activities.

Infrastructure, information formats, software tools, interoperability technologies.

In an industrial framework, structuring business knowledge in an information processing system does not imply facilitation of communication with another business system. Data interpretation changes according to the business and the challenge is in the ability to preserve information semantics when communicating.

Building interoperability architecture for communication can align Business, Knowledge and ICT through semantic framework to ensure compliance when exchanging data. In the following section, we will explain a deployment of the interoperability framework to present a communication architecture adapted to our context.

3.2 General Requirement for Interoperability

Product data are compiled from various functional divisions which interact between each others for the creation and manufacture of the product. Each of the vaccine supply chain divisions contributes by creating different types of data and information.

From one stage to another, product data are recorded using a specific structuring and format. At each division, the information system is defined in accordance with business context needs. The product in vaccine industry is not tied to physical shape except in this packaging stage. Thus, there are a particular need for some particular information systems to support product data, especially in his design stage.

The major quest for each vaccine company is product quality. This objective is achieved only by ensuring a better communication at these different levels: business, knowledge, applications and data. The interoperability concepts aim to provide some solutions to communicate heterogeneous information systems involving in different business context.

In the perspective to ensure data compliance of the production data according to their definition in the Marketing Authorizations (MA) documents, we propose hereunder the means to use the MA data, which can be read only by pharmacists, to adapt them to logisticians needs. The used approach makes it possible to ensure interoperability between the supporting information systems, while satisfying some business constraints.

4 Interoperability for Production Data Compliance

In our context, the objective behind the establishment of the communication between the information systems is to ensure the conformity of the product data in one system in relation to each other. Based on the description of information in an Marketing Authorization, it is necessary to return the product data values, useful for the production, to the ERP.

4.1 From MA to ERP

As we mentioned before, the following systems are involved in our context:

Marketing Authorization (MA) information system: generally managed by the regulatory affairs division of the company and constitutes a collection of product information. A MA is composed of electronic documents coming from several sources and contains, for example, scanned documents, reports related to research and development activities and other documents. The semantic structuring of these authorizations documents provides a format and content harmonized according to a pharmaceutical vision following the Common Technical Document (CTD) format defined by the International Conference of Harmonisation [13] [14]. Even using

CTD format, it is not always easy to find all the information needed for production process. Some Biologic and pharmaceutical backgrounds are necessary to find the needed information from regulatory data. Also, with a very large number of MA documents – that's run into thousands of pages – it is very difficult to find all production information. The MA represents regulatory aspect of product data.

ERP (Enterprise Resource Planning) systems: as an Enterprise Inwards Solution, ERP systems [15, 16] cover an intra-enterprise integration of various functions based on MRP II concepts with the added functionalities of finance, distribution and human resources development, integrated to handle the global business needs of a networked enterprise. The ERP covers different divisions of the company and regroups complex modules and functionalities. The production planning module use the technical definition of the product to generate new dynamic data, such as working orders. When the ERP data are non-conform to the right product data definition in the regulatory documents (MA), generated products are necessarily non compliant.

Each division presents a specific vision of the product with local knowledge tied to its business needs. To ensure the conformity at the product data definition level during its translation (from the regulatory systems to the ERP), it is necessary to define a communication platform to include the different viewpoints: organizational, business, informational, and technical [17].

4.2 Type of Data to Be Translated

The product structure is defined in both MA and ERP systems as a specific series of "product states". The biologic description of the product and its various states related to the manufacturing phase are presented in a CTD part called "product quality". These states are not necessarily coherent with the actual production states. To guaranty the coherence of product data, it is suited to organize these data according to the product states. However, the problem still concerns the conformance of data values for each product states during the translation process. We should take care about the definition of these states and data semantic in each one. For example the shelf life of an intermediate product substance (state) is 3 years, at a storage temperature of -70°C if it was preserved with no alteration (as is) and 1 year if it was stored at 5 °C.

In the manufacturing phase, we assume that the product has a fixed number of states (reflected into the information system). It is necessary to identify from the ERP and the regulatory information system the entire specification of each state. This is achieved by what we call "product states reference frame". The reference frame represents the structuring of one product datum that assigns for each product state, the data value, rules applied to extract data from the information system, and business constraints helping to understand the choice of data value. For each product state, we need to define also some components of the bill of materials of this state. For example, when our final product is presented (at its final state) in the form of two substances: powder and excipient (i.e a substance combined with a drug to make it easier to administer), we need to specify shelf life for these two substances.

The application of this reference frame to product data consists in seeking data values of all states in accordance to rules and business constraints already identified. Fig. 4 illustrates this structuring.

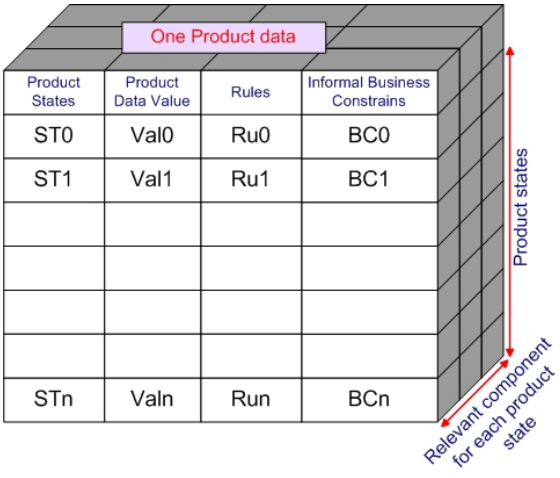


Fig. 4. Product states reference

This reference frame represents the data profile in both information systems. It must be updated during a potential modification of the structure of the product. It can also be published in the organization to ensure better comprehension and exploitation of the product data.

Each line of this reference frame contains the product state components and for each one of them, the value to be validated, the rules which allow to extract and transform data and business constraints. The interoperability process is supported by the link between these values, rules and constraints.

4.3 Rules Definition

The definition of the rules is a tedious phase and requires three levels:

4.3.1 Production Information Rules

These rules specify when to extract or to insert data into the ERP. Some difficulties arise when attempting to insert data because ERPs are characterized by the re-use of components (product states information). Taking a close look into two vaccine products, there is a great probability to find the same component. In this case, there are invariably one or more specific common production states with the same coding in the system.

In the ERP, and following a request for modification of a data value for a product state, it is necessary to check if the reference for this state is already used by another product. Considering the complexity of the ERP architecture and the overlapping between the product states information, it is difficult to seek products by a simple

indication of an "intermediate" state. For example, such identification can take up to two days to find all concerned product states and theirs dependencies. If we schematize product states by a tree structure, the overlapping between branches can be possible everywhere except at the top level (tree leaves). Fig. 5 shows an example of these overlapping. Each product has 6 states: S1 to S6.

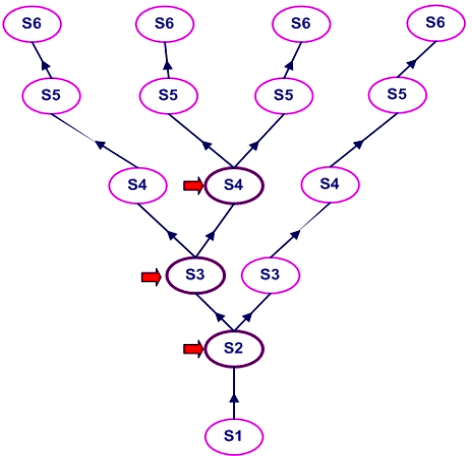


Fig. 5. Overlapping between product states for different products in the ERP system

Integration rules are used to control the existence of any overlapping between the tree branches (resulting in common states) as well as the impact of any data modification on the product structure and its states. The impact of some modifications or transformations at the data level is sometimes governed by informal business constraints. For example, the manufacturing date of a product is notified as the starting date of the first valid test of stability. If we want to change the shelf life of a state, the expiry date must be revalidated. This aspect is important for data understanding. This is why we added informal business constraints to each product state. Moreover, these constraints help understanding the context in which the product states rules are used, and in mapping the ERP "product states reference frame" to its corresponding reference frame in the MA information system.

4.3.2 Mapping Rules

These rules are defined for mapping process between "product states reference frames" by establishing links between "active product states". From all predefined product states in one reference frame, active product states present significant states with data value. Performing these links present a regulatory and pharmaceutical responsibility that is necessary to share with production, to ensure the coherence of rules. The product states are not the same across information systems and across reference frames. From one product to another, a state may or may not exist. We use different business knowledge as references to create these links of communication between active states. We notify these information on both MA and production reference frames (ERP).

The mapping rules allow the formalization of the data fields to be inter-connected (links **n** .. **n**). Active states data values in regulatory reference frame generate corresponding values in the product states reference frame of the ERP. Fig. 6 illustrates examples of connection modes. One state in each reference frame can correspond to one or more states in the second and vice versa. To generate mapping rules, we should analyze data and rules from the two reference frames. For example, mapping rules could be the adding of states data values, the calculation of their average or their minimum, etc.

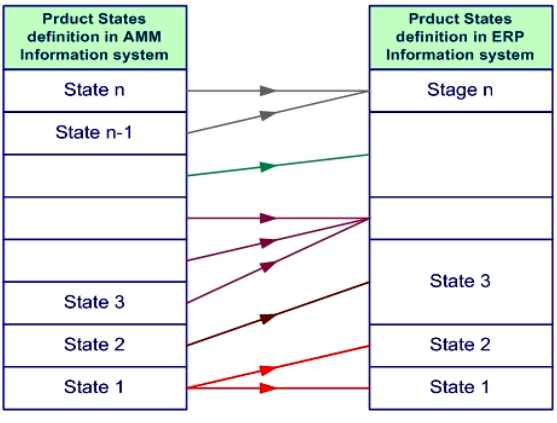


Fig. 6. Mapping links

4.3.3 Regulatory Information Systems Rules

According to vaccine data structuring, the information system which manages the MA is not able to be directly interfaced to the regulatory product states reference frame. It is possible to have several MAs for only one product, and conversely, one MA for several products. These characteristics are relocated on product states, which increase the complexity of the information retrieval. It is frequent to find for example two product authorizations with various destinations (country) or presentations (filling containers) and having a common product state but with different data values. This difference is due to the history of the negotiations between the company and health authority about the MA content.

In the following section, we will explain the need for defining different rules types and later (in a future work), we will present, through a multi level modeling approach, different kinds of rules that we need to create.

5 Case Study

5.1 The Compliance Scenario

This case study presents an illustration of an application developed within Sanofi Pasteur Company, a firm specialized on the biologic development and the production

of vaccines for human use. The purpose of this application is to ensure compliance, from the MA to the ERP, for these three data: *Site of Manufacturing*, *Shelf Life*, and *Storage Condition*.

All MA data were extracted from e-TRAC (Electronic Tracking of Registrations and Commitments), an information system developed to track MA evolutions. Access to these data is ensured through web interface allowing us to export a predefined reports from RA-Cockpit reporting module. As presented in fig. 7, we can:

a) export data for one product line to create the report ,

b) distribute this report by product licence number as criteria to identify different product data,

c) for one product data, instantiate three reference frames for regulatory product states,

d) apply mapping rules to generate corresponding ERP (here SAP) product states reference frame,

e) use the same specific criteria for data structuring in SAP to validate data (comparing to those coming from SAP reference frame generated after mapping).

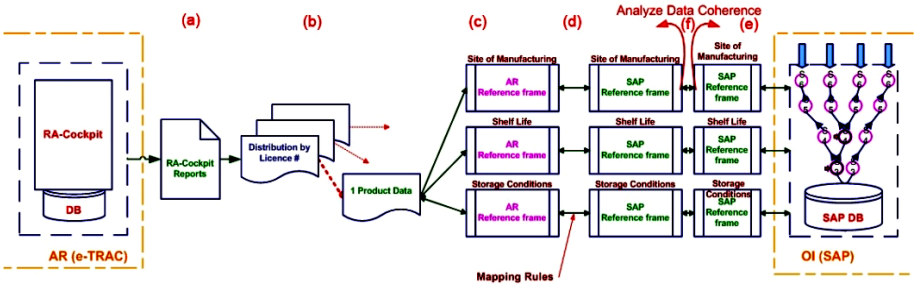


Fig. 7. Communication scenario

5.2 Validate Data in Sap

As mentioned before, there is a great probability to have the same product state in different product states decompositions. So, we can find the same value for the same product state in different SAP reference frames. In SAP system, we identify each entity, called item, by one reference. That is why, in addition to the first SAP reference frame generated after mapping, we instantiated a second SAP reference frame with only SAP code and corresponding data value field. In this second reference frame it is necessary to find, from SAP, the code and value of each product state. Due to specific information structuring in SAP at Sanofi-Pasteur Company, we can find the item code for the last state (final product) and use "item code filiations" (Where-Used technology) to find the code for previous product states and their data values starting from the last.

Actually we have two SAP reference frames: one with data values generated after mapping process from the regulatory reference frame, and the second with data values and item code coming from SAP. We define here some new rules of coherence:

R1: For the same product state, there is necessarily the same data value, otherwise notify a compliance exception,

R2: The same item codes in the second SAP reference frames (corresponding to different products) should have the same associated data values, otherwise notify a compliance exception,

It is frequent to find two or more MAs or registrations that differ just by product name from one country to another. For example we can define influenza (Flu) vaccines for entire Europe, but during the structuring of the product information in the e-TRAC, we should separate the products by country.

R3: Validating the three data (Site of Manufacturing, Shelf Life, Storage Condition) for grippe in a particular region, requires the same data values in e-TRAC reference frame for all countries of this region, otherwise notify a compliance exception.

Finally, within this Flu line product vaccines case study, the applied architecture and its rules provided an interesting solution by ensuring compliance of 94,6% of the final products for the used three data: *Site of Manufacturing*, *Shelf Life*, and *Storage Condition*. One of the reasons of non-total compliance is related to the existence of quality level information in the MA system that has no correspondence in the ERP system.

6 Conclusion

In this paper, we presented a data exchange methodology between heterogeneous information systems. We particularly focused on product structuring and explained dependencies between product data in the vaccine field. Our main objective is to ensure data compliance between two information systems, one related to the Marketing Authorizations and the other related to production, through the establishment of communication architecture. We based our work on the mapping between product "states" information along the product manufacturing life cycle. In spite of differences in their business visions, both systems use the product manufacturing decomposition as guide-line for structuring the information.

Our methodology treats only the information coming from Marketing Authorizations systems to map and validate it in the ERP systems. However it does not treat product information that exists in the ERP systems and is not related to any MA system. The next step of this work will focus on the generalization of the used rules and constraints, not only to extract or integrate data through reference frames, but also between product states in a same reference frame.

Acknowledgements

This work is elaborated with the collaboration of Sanofi Pasteur France company where Mr. Néjib MOALLA is integrated as a researcher in Industrial Operations services during his PhD thesis (CIFRE N° 865/2005).

References

1. U.S_FDA: Pharmaceutical CGMPS For The 21st Century — A Risk-Based Approach Final Report. Department of Health and Human Services, 32 (2004)
2. US_FDA: Challenge and Opportunity on the Critical Path to New Medical Products. U.S. Department of Health and Human Services, 38 (2004)
3. Salinsky, C.W.E.: The vaccine industry: Does it need a shot in the arm?: Background Paper. National Health Policy Forum, 34 (2006)
4. Grace, C.: Global health partnership impact on commodity pricing and security. DFID Health Resource, 22 (2004)
5. Thaul, S.: Vaccine policy issues for the 108h congress. The library of Congress, 16 (2003)
6. Schweigera, R., B.M., Hoelzerc, S., Dudecka, J.: Health care systems using XML Standards. International Journal of Medical Informatics 74, 267–277 (2005)
7. Schweigera, R., B.M., Hoelzerc, S., Dudecka, J.: Linking clinical data using XML topic maps. International Journal of Artificial Intelligence in Medicine 28, 105–115 (2003)
8. EbXML: Technical Architecture Specification. ebXML Technical Architecture Project Team, 39 (2001)
9. Sang Bong, K.Y.Y.: Web-based knowledge management for sharing product data in virtual enterprises. International Journal of Production Economics 75, 173–183 (2002)
10. William, C.B.: Product data markup language: a new paradigm for product data exchange and integration. International Journal of Computer-Aided Design 33, 489–500 (2001)
11. NoE_ATHENA: Second Version of State of the Art in Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability, p. 212 (2005)
12. NoE_ATHENA: Interoperability Roadmap up date, p. 30 (2005)
13. ICH: COMMON TECHNICAL DOCUMENT. In: International Conference On Harmonization Of Technical Requirements For Registration Of Pharmaceuticals For Human Use, p. 136 (1999)
14. ICH6: New Horizons and Future Challenges, Summary Report. In: Sixth International Conference on Harmonisation, Osaka, Japan, pp. 13–15 (2003)
15. Martin, V.: Software component architecture in supply chain management. International Journal of Computers in Industry 53, 165–178 (2004)
16. Kelle, A.A.P.: The role of ERP tools in supply chain information sharing, cooperation, and cost optimization. International Journal of Production Economics, 41–52 (2005)
17. Gao, X., H.A., Maropoulos, P.G., Cheung, W.M.: Application of product data management technologies for enterprise integration, vol. 16, pp. 491–500. Taylor & Francis, London (2003)

Evaluating Server Capacity for Streaming Media Services

Beomjoo Seo[1], Michele Covell[2], Mirjana Spasojevic[2], Sumit Roy[2], Roger Zimmermann[3], Leonidas Kontothanassis[2], and Nina Bhatti[2]

[1] University of Southern California, Los Angeles, CA 90089, USA
[2] Hewlett–Packard Laboratories, Palo Alto, CA 94304, USA
[3] National University of Singapore, Singapore 117543

Abstract. The recent proliferation of streaming media systems in both wired and wireless networks challenges the network operators to provide cost-effective streaming solutions that maximize the usage of their infrastructure while maintaining adequate service quality. Some of these goals conflict and motivate the development of precise and accurate models that predict the system states under extremely diverse workloads on-the-fly. However, many earlier studies have derived models and subsequent simulations that are well-suited only for a controlled environment, and hence explain a limited sets of behavioral singularities observed from software component profiles. In this study, we propose a more general, procedural methodology that characterizes a single system's streaming capacity and derives a prediction model that is applicable for any type of workload imposed on the measured system. we describe a systematic performance evaluation methodology for streaming media systems that starts with the reliable collection of performance data, presents a mechanism to calibrate the data for later use during the modeling phase, and finally examines the prediction power and the limitations of the calibrated data itself. We validate our method with two widely used streaming media systems and the results indicate an excellent match of the modelled data with the actual system measurements.

1 Introduction

The recent developments in media compression technologies such as MPEG-4 and the tremendous growth in available end-user network bandwidth in combination with infrastructure-level services such as Content Delivery Networks (CDN) have made streaming media an ubiquitous web application. As streaming media becomes an increasingly important part of the data traffic, there is a growing need to characterize server behavior and understand end–user's streaming experience, expecting to make the best use of streaming infrastructure.

Traditionally, server performance has been observed by examining simple metrics such as CPU, disk, and network utilization. However, such singular metrics do not capture the complex interdependence of resources and may result in either under- or over-provisioning of the infrastructure. In this study, we propose a systematic and exhaustive methodology for evaluating the performance of streaming media services, utilizing both server- and client-side measurements under limited, but representative streaming workloads.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 112–131, 2008.

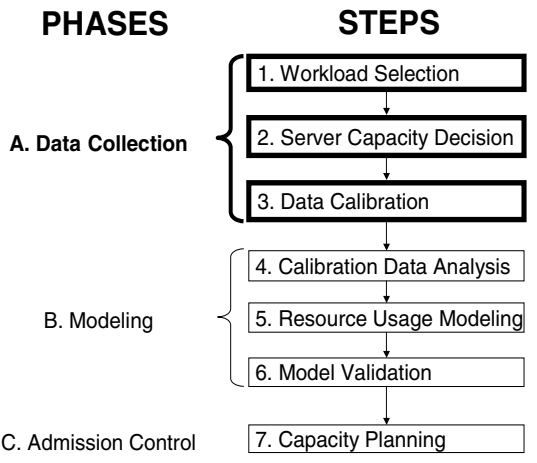


Fig. 1. A procedural methodology to characterize streaming media servers

Figure 1 outlines our multi-stage server characterization process that aims to estimate current server load accurately and reliably. The process consists of three main phases with a total of seven sub-steps. The goal of the *data collection* phase (Fig. 1 A.) is to collect meaningful, usable performance data, and consists of three procedural steps: (1) identifying a set of disjoint workloads, (2) collecting and deciding server capacity for every disjoint workload, and (3) calibrating the collected performance statistics. The first step – *workload selection* (Fig. 1 A.1.) – is a design process to define a set of disjoint workloads with precise properties. The proper choice of non-overlapping workloads reduces the scope of experiments. The second step, *server capacity decision* (Fig. 1 A.2.), is the most crucial in the collection phase, since it defines the maximum number of concurrent users that are admitted to the system while provided with acceptable service quality. We term it "server capacity" or "saturation point". The high complexity of the interactions among system components introduces high statistical variations, which lead to irregular service failure near a saturation point. Hence, we propose a rigorous saturation decision model that runs experiments until the experimental results provide a convincible decision. After the decision, the last step, *data calibration* (Fig. 1 A.3.), recollects and labels resource usage profiles.

After the data collection phase, a careful analysis of the calibrated data (Fig. 1 B.) identifies which client and server resources are the dominating factors that contribute to the measured system saturation. In the final phase, the identified resource usage may provide the basis for off–line capacity planning or even for online admission control (Fig. 1 C.).

In this article, we focus on the design of the performance evaluation methodology that is suggested for the first phase. While our prediction model is not the main theme of this article, we will briefly introduce the model. Its detailed mathematical derivations are covered in [1].

The rest of this paper is organized as follows. Section 2 describes our performance data evaluation methodology. In Section 3 we validate our methodology by extensively

measuring the performance of existing streaming servers. The prediction model and its validation results are briefly summarized in Section 4. Section 5 presents the related work. Finally, we conclude and present ideas for future work in Section 6.

2 Methodology

This section presents the evaluation methodology for measured performance data. Section 2.1 describes the types of client workloads on which we measure. In Section 2.2, we define the notion of service failures and explain the decision criteria for server saturation.

2.1 Workload Selection

Synthesizing workloads that were derived from observations is widely used in simulated environments. While such generalized workload model helps to understand the average behavior of a streaming server, it usually fails to detect whether the server is saturated. Sometimes, server resources such as processing power, I/O bandwidth, memory size, data layout, or various bitrates of encoded content may be overloaded temporarily, which, unfortunately, may cause the system non–operational for a while. Thus, it is impractical or even impossible to distinguish all the workloads that reflect every possible streaming behavior.

Our approach for choosing a set of workloads for our benchmark experiments and evaluations is to represent the complex and large streaming workload space with a number of non–overlapping sets – we term such non–overlapping workloads **pure workloads**. To narrow the evaluation scope to a practical number of experiments and still maintain the rich expressiveness of a general workload, we classify the pure workloads along three dimensions: source location of a media content, access popularity, and encoding rate.

Source location. A server may retrieve the data from its local storage device or from another remote server. We refer to the former as a **VoD** session and to the latter as a **Live** session.

Access popularity. The access frequency of an opened file varies from a single client access to a large number of client accesses. Due to the many possible combinations of different access frequencies, we focus on two types: popular and unpopular. **Popular** content is a single source that is viewed by all client sessions. We refer to content that is accessed only by a single client as **Unpopular**.

Encoding rate. The encoding bitrate of the content is also a continuously varying dimension. For our calibration purposes, we select two typical bitrates: 300 kbps **High-rate** source material and 78 kbps **Low-rate** material.

We use the following abbreviated naming scheme: the first letter defines the source location (**L**ive or **V**oD), the second letter defines the access popularity (**P**opular or **U**npopular), and the third letter defines the bit-rate (**H**igh-rate or **L**ow-rate). For brevity, we replace the third letter with x to signify both high and low bit rates when appropriate. Table 1 summarizes these conventions.

Table 1. Pure workload matrix and its naming convention

	Popular		Unpopular	
	High Rate	Low Rate	High Rate	Low Rate
VoD	VPH	VPL	VUH	VUL
Live	LPH	LPL	LUH	LUL

2.2 Server Capacity Decision

When a server approaches overloaded, it may start to perform erratically – that is, request rejection or abnormal session termination. Such failures are, however, apt to occur even at lower loads, since operational spikes may happen at any time. Thus, determining the maximum number of concurrent streaming sessions per pure workload consistently and reproducibly becomes very challenging only by observing from simple server statistics. For a reliable server capacity decision, we examine the performance data collected not only from a server but also from clients.

Client Logging. We have developed a light–weight client application that requests an RTP [2] stream to a media server, accepts RTP packets, and records session-level statistics. In addition, it can record a trace of every RTP/RTCP packet – i.e., packet arrival time, packet size, sequence number, and the media decode time.

Every experiment runs two types of client applications: *loading clients* and *probing clients*. A loading client is a long-lived session that exercises the server at the level of concurrent requests. To support a very large number of simultaneous loading clients (nearly 3000 clients in some situations), it only records session-level statistics. The probing client is a short-lived session that is issued consecutively to collect detailed session statistics after the experiment launches all the loading clients and reaches steady state. It records both session-level statistics and a trace of the delivered data packets. Thus, each experiment runs a specified number of loading clients during the experimental period and a probing client intermittedly.

From the trace, we can also derive the number of rebuffering events, which is the number of late-arriving packets observed from the probing session. Late-arriving packets are computed from the packet-arrival offset, the difference between each packet delivery time and its expected deadline. This offset value is greater than zero when the packet is late and less than zero when it is early. After the detection of a first late-arriving packet, deadlines of the remaining packets are shifted accordingly. Otherwise, subsequent packets might be detected as a late, incurring an unnecessary rebuffering penalty. The detection of rebuffering events was, however, often problematic due to increasingly bursty packet transmissions as the server workload increased. The timing of these bursts was such that, on occasion one or two packets would be delayed beyond their delivery deadline. This small amount of over-delayed data resulted in rebuffering violations on those experiments, even when the server was otherwise not saturated. We found that, by re-categorizing these few packets as being lost data (instead of late data), we could avoid a rebuffering violation without inducing a size violation. This greatly improved the reliability and reproducibility of our decision surface.

Failure Detection. If the server system is overloaded, a newly delivered streaming request may be either rejected or admitted but experiencing degraded session quality. Among session failures, some can be detected from error log files easily (hard failure), while others need further processing (soft failure). Admission rejection and explicit session termination in the middle are hard failures.

Soft failure is a general term that describes an unacceptable user streaming experience. Duration violations, size violations, and rebuffering violations belong to this category. These are defined as follows:

- **Duration Violation:** A session s that satisfies the following inequality condition

$$|\frac{T(s)}{T_s} - 1| > \rho_T \ ,$$

  where T_s is the expected duration, $T(s)$ is the measured duration, and ρ_T $(0 < \rho_T < 1)$ is the acceptable range of the duration, is considered to violate the duration requirement.
- **Size Violation:** A session s that satisfies that following inequality condition

$$1 - \frac{B(s)}{B_s} > \rho_B \text{ under } B(s) < B_s \ ,$$

  where B_s is the expected amount of data bytes received at the client, $B(s)$ is the measured size, and ρ_B $(0 < \rho_B < 1)$ is the acceptable range of the bitstream length, is considered to violate the session length requirement.
- **Rebuffering Violation:** Any experiment which has N number of individual probing statistics and satisfies following inequality condition

$$\frac{\sum_s^N \{I(s) + P \cdot R(s)\}}{\sum_s^N T_s} > \rho_Q \ ,$$

  where $I(s)$ is the start-up delay of a measured session s, $R(s)$ is the sum of time periods when s was in a rebuffering state, P is the penalty constant assigned per rebuffering event, and ρ_Q $(0 < \rho_Q < 1)$ is the acceptable range of the service quality, is considered to violate the desired service quality.

Duration and size violations are obtainable from session-level statistics, while rebuffering violations are computed from data packet traces available at client log statistics. Our failure model excludes the condition $B(s) > B_s$, where the test session receives more packets than expected, which is caused by packet retransmission. In slow networks, excessive retransmissions might not meet the network bandwidth requirement, which may well be considered a failure. In our experiments however, as long as service quality is acceptable, we consider this case as acceptable.

To evaluate user's experience, we may directly measure the quality of voice samples and the quality of video images received at the client side [3,4] or indirectly estimate a user's frustration rate. We prefer the less accurate but real–time quality evaluation method. Otherwise, the server capacity decision would take a tremendous amount of time to finalize due to its stepwise nature. For this reason, we chose Keynote's indirect

Algorithm 1. FM: detect any failure from client logs

Require: $\mathbb{L}$: a set of statistics of loading clients
$\mathbb{P}$: a set of log statistics of probing clients
1: {examine all client logs whether there is any hard failure, duration violation, or size violation.}
2: **for all** $s \in \mathbb{L} \cup \mathbb{P}$ **do**
3: **if** $T(s)$ is null or $|\frac{T(s)}{T_s} - 1| > \rho_T$ or $1 - \frac{B(s)}{B_s} > \rho_B$ where $B(s) < B_s$ **then**
4: return failure
5: **end if**
6: **end for**
7:
8: {detect rebuffering violation from probing statistics.}
9: $frustration \leftarrow 0$
10: **for all** $s \in \mathbb{P}$ **do**
11: $I \leftarrow$ startup delay of s
12: $R \leftarrow$ number of rebuffering events of s
13: $frustration+ = (I + \text{Penalty} \cdot R)$
14: **end for**
15: **if** $frustration > (\rho_Q \cdot \sum_{s \in \mathbb{P}} T_s)$ **then**
16: return failure
17: **end if**
18: return success

method [5]. The frustration rate proposed by Keynote Inc. is a well-established methodology to quantify a user's streaming experience. This measure computes the waiting time spent at startup, initial buffering, and rebuffering events of a measured session. To minimize false negatives caused by statistically generated spikes during the experiments, our methodology extends Keynote's rating system by collecting and analyzing multiple probing sessions.

Algorithm FM states our failure detection model formally. In line 3, the case '$T(s)$ is null' means that the measured duration of s is not recorded. It may be caused by service denial, abnormal session termination, or no log because of no response from the server. After detecting all hard failures and duration/size violation, lines 9–17 detect rebuffering violations from probing session statistics.

If any session failures are seen at any time during the experimental epoch, the streaming server is labelled as being saturated for the full experimental epoch. Each experimental epoch used to determine the saturation point consists of five 20-minute measurement sets at a possible saturating workload. This repetition ensures a reproducible and consistent categorization of the server.

Server Saturation Detection. The term **saturation** that describes a server's overloaded state is confined as follows: *for a given workload, server will be determined saturated if the Algorithm* `IsSaturated` *returns* $saturated$. Accordingly, the server capacity is defined as the maximum number of loading sessions that can be executed while the server is not saturated.

Algorithm `IsSaturated` repeats the experiments x times by running n number of w-type pure workload sessions.Among the repeated experiments, if only a single

Algorithm 2. `IsSaturated`: validate whether a given experiment is saturated

Require: w: pure workload type,
n : number of client sessions to run,
x : number of repetitions
1: **for** $i \leftarrow 1$ to x **do**
2: $\mathbb{L}_i, \mathbb{P}_i \leftarrow$ perform an w-type experiment(n)
3: **end for**
4: **if** $\exists\ J \leftarrow \{j|$ **FM(**$\mathbb{L}_j, \mathbb{P}_j$**)** $=$ failure $\}$ **then**
5: **if** $|J| = 1 \cap$ **FM(**$\mathbb{L}_j, \mathbb{P}_j$**)** reports hard failure **then**
6: {examine again.}
7: **for** $i \leftarrow 1$ to x **do**
8: $\mathbb{L}, \mathbb{P} \leftarrow$ perform a w-type experiment(n)
9: **if FM(**$\mathbb{L}, \mathbb{P}$**)** = failure **then**
10: return $saturated$
11: **end if**
12: **end for**
13: return $not\_saturated$
14: **end if**
15: return $saturated$
16: **end if**
17: return $not\_saturated$

hard failure is reported (lines 4–5), the algorithm repeats the experiments (lines 7–8) in order to check whether any performance spike occurred. This is necessary because performance spike usually leads to a hard failure not a soft failure. During the second repetition (lines 7–8), any occurrence of any type of failure report (line 9) finalizes the decision that the given system is saturated.

Using the Algorithms `FM` and `IsSaturated`, the Algorithm `SC` automates the server capacity decision. For each pure workload (line 1), it first finds a possible saturation load by scanning the test region exponentially (lines 4–7). If any failure is reported for the given load, the algorithm finalizes the saturation decision by repeating the experiments five times (line 7). If the decision were being made saturated, we would have found the upper bound of saturation region. To find the lower bound of unsaturated region, we execute the validation procedure (lines 10–12) for the load ($\frac{n}{p}$) which was previously reported unsaturated. After deciding the upper bound and the lower bound of the further testing region, the algorithm iterates through the range with a binary search until it finds out more accurate unsaturated load (lines 15–26). Line 18 is the condition that guarantees to terminate the iteration. The last step (lines 28–37) is to re-assure that our server capacity decision is correct by observing nearby regions. In our algorithm, we limit the observation scope by examining the load of 97%, 98%, 99% of the candidate server capacity (lines 29–30).

2.3 Data Calibration

To derive an accurate prediction model under a classic labelled training data approach, we introduce calibration measurements that associate the measurement data and the

Algorithm 3. SC: Server Capacity Decision

Require: $\mathbb{W}$: a set of pure workloads
p : carefully chosen exponent number
for all $w \in W$ **do**
  {scan saturated region exponentially.}
  $n \leftarrow 1$
  repeat
    $n \leftarrow n \times p$
    $\mathbb{L}, \mathbb{P} \leftarrow$ perform a w-typed experiment(n)
  until FM($\mathbb{L}, \mathbb{P}$) = failure $\cap$ **IsSaturated**($w, n, 5$) = saturated

  {verify min is not saturated while max is saturated.}
  while IsSaturated($w, \frac{n}{p}, 5$) = saturated **do**
    $n \leftarrow \frac{n}{p}$
  end while

  {binary search based server capacity decision.}
  $min \leftarrow \frac{n}{p}, max \leftarrow n$
  loop
    $n \leftarrow \lfloor (min + max)/2 \rfloor$
    if $n \leq min$ **then**
      exit infinite loop
    end if
    if IsSaturated($w, n, 5$) = saturated **then**
      $max \leftarrow n$
    else
      $min \leftarrow n$
    end if
  end loop

  RETRY: {verify final decision by examining nearby regions.}
  for $i \leftarrow 1$ to 3 **do**
    $l \leftarrow min - \lceil (min \cdot i \cdot 0.01) \rceil$
    $\mathbb{L}, \mathbb{P} \leftarrow$ perform an w-typed experiment(l)
    if FM($\mathbb{L}, \mathbb{P}$) = failure $\cap$
    IsSaturated($w, l, 5$) = saturated **then**
      $min \leftarrow l$
      goto RETRY
    end if
  end for
  w.ServerCapacity $\leftarrow min$
end for

label data. Training and testing on the same experimental configuration does not provide a realistic measure of the robustness of the derived models. We believe that we avoid that unrealistic optimism in this article by training and testing on distinct configurations of workloads.

Once we have found the server capacity for the given set of pure workloads, we collect measurement data at pure workloads below that capacity. Using only pure-workload calibration data allows us to minimize the required number of experiments: the server administrator who uses our approach does not need to recreate all of the different workload mixtures under which the server will operate.

Since we are primarily interested in recognizing server saturation and in predicting the transition from unsaturated to saturated states under additional workloads, we collect this measurement data in the range from 70% to 100% saturated. We collected one 20 minute measurement period on 70%, 75%, 80%, 85%, 90%, 95%, and 100% of the saturating loads for each of the pure workloads.

During the measurement period, we collect the server–side resource usage statistics as well as client–side loading/probing statistics. Server performance metrics for the duration of each experiment are collected on the server itself using statistics collected by the Linux kernel and reported under the */proc* filesystem.

3 Evaluation Results

In this section, we discuss the results that we observed when calibrating the Apple *Darwin Streaming Server [6]* and the RealNetworks *Helix Universal servers [7]*. While sharing the similar core architecture, they use very different internal policies, leading to different performances even with the same hardware. While a single specific performance model does not need to handle both servers, the evaluation methodology should be independent of the particular server software.

We use box–and–whiskers plots: the horizontal line in the middle of the box is the median value; the lower and upper lines of the box are the 25th and 75th percentile of the sample data.

3.1 Experimental Setup

Our experiments run on three distinct sets of machines: the streaming-server machine that is being calibrated or tested; up to four live-source machines; and up to six client machines. The server machine is a dual 1.4GHz Pentium III PC with 1GB memory, running SuSE 8.2 (kernel version 2.4.20). The other machines are selected to have sufficient computation and I/O capacity[1], so that they do not influence the experimental results. All the machines were connected to a switched Gigabit network, and isolated to avoid uncontrolled network interference. The streaming-server software suites used for this article are the *Darwin Streaming Server, v4.1.3*, and *Helix Universal Server, v9.0.3.916*.

To avoid performance variations, we used multiple distinct copies of the same material for Live and VoD tests. For our Live tests, the material was stored on the live-source machines and was relayed through the streaming server, using the *Darwin Playlist-Broadcaster* [6]. To minimize performance variations due to interfering disk accesses, measurement logging on the streaming server was done to a separate disk.

[1] In our test-bed, the live–source and client machines have 1.0 - 2.4 GHz Pentium III processors with 256 MB - 1GB memory.

Each experimental period has three distinct phases: ramping up, steady-state, and termination. During the first phase, loading clients are added at 500 ms intervals, which avoids start-up failures purely due to transient effects. The loading clients are used to induce a particular type of workload on the server. After reaching the steady-state period, we collect measurements from the streaming server machine. We also sequentially launch 20 probing clients, which run for non-overlapping 1 minute periods. Since probing-client statistics will also be collected on in-service streaming–media servers, we want the probing client to be both informative and low-overhead. For this reason, we use a VUL request for each probing session. We chose a VoD request since we could always be assured that the chosen file-based content would be available and unchanging from one probe to the next. We chose Unpopular since we do not always know what content is currently Popular (that is, likely to be partially in the file–buffer cache) but we can request distinct probe–only content and be assured that it is unpopular. We chose Low-rate to minimize the overhead induced by the probe.

3.2 Maximum Server Capacity

Tables 2 shows the results of the finalized server capacities measured with three different experimental setups. The first set (1) Darwin was performed in the Darwin environment without any systematic decision model. The server saturation decision was heavily relied upon experts' intuition. This approach detected some hard failures, duration violations, and size violations. The second and the third sets were executed with the Darwin and the Helix experimental setup, respectively, using our proposed methodology. In our method, we used 0.03 for ρ_T, ρ_B, ρ_Q, or 3% allowances. To validate the correctness of our server capacity decisions, we intentionally collected performance data not only from near the saturation area but also from various other areas.

Table 2. Different sets of experimental setups and their server capacity

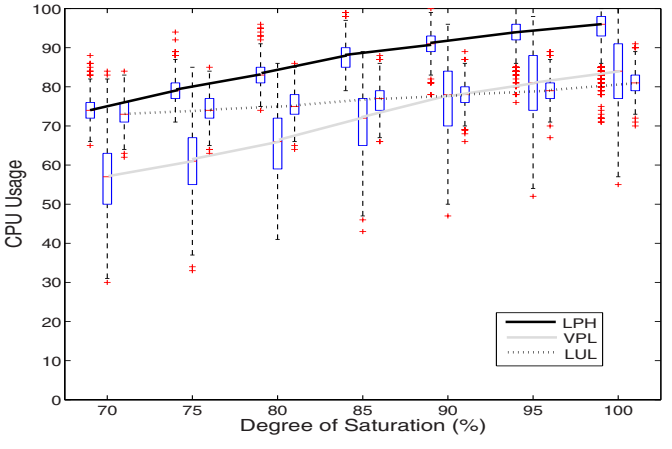
	VPH	VPL	VUH	VUL	LPH	LPL	LUH	LUL
(1) Darwin[a]	438	780	36[b]	170	996	–[c]	–[c]	405
(2) Darwin[d]	425	726	33[b]	259	1158	1976	405	405
(3) Helix	590	1220	91	228	1460	2870	396	492

[a] Intuitive approach.
[b] Excessive data copy thrashing between user space and kernel space was reported.
[c] Experiments failed due to system misconfiguration.
[d] With our saturation model.

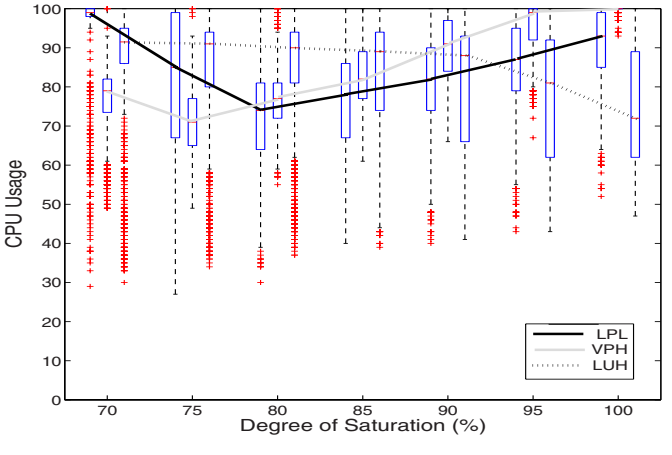
With the (2)Darwin experiments, several pure workloads exhibited a different failure reason for the server to saturate. For example, the failure type of the VPH workload that was determined due to hard failure on the (1)Darwin set later turned out to be a rebuffering violation. This inconsistency was caused by the existence of a rebuffering violation which had occurred before the system experienced a hard failure. The dramatic server capacity change on the Darwin server sets (52% difference for the VUL experiments)

is largely due to improper handling of temporary performance spikes. Even when the (1)Darwin measurements were repeated around what was thought to be the saturation points, temporary performance spikes inhibited the right decisions. One strange behavior was reported during the Darwin LUx experiments where we couldn't collect any session statistics because all the sessions were unable to play their streaming request throughout the whole experiment epoch whenever we loaded more than 405 sessions. The Helix server, however, could load more than 405 sessions.

When comparing different servers, we found that the Helix server achieved higher system throughput than the Darwin server for the CPU–intensive workloads such as VPx and LPx. For I/O intensive workloads, the Darwin server reported a slightly improved throughput for VUL and LUH. As noted in the footnote of the Tab. 2, the Darwin

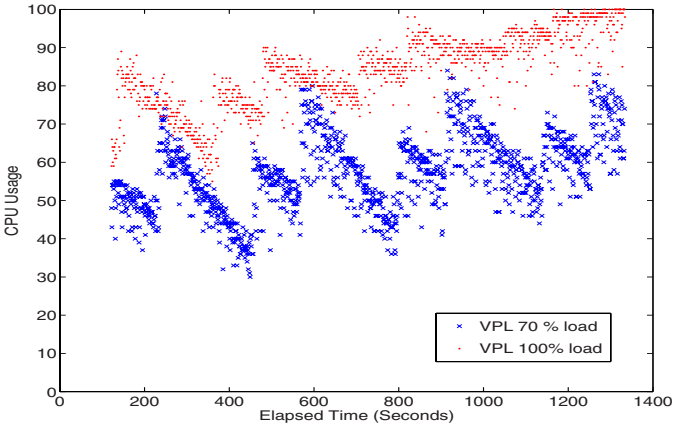


(a) Darwin

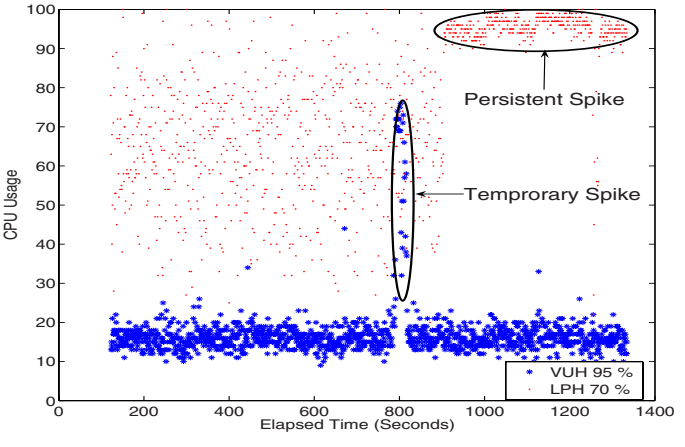


(b) Helix

Fig. 2. CPU usage as a function of the degree of saturation



(a) Darwin: sawtoothed temporal dependency



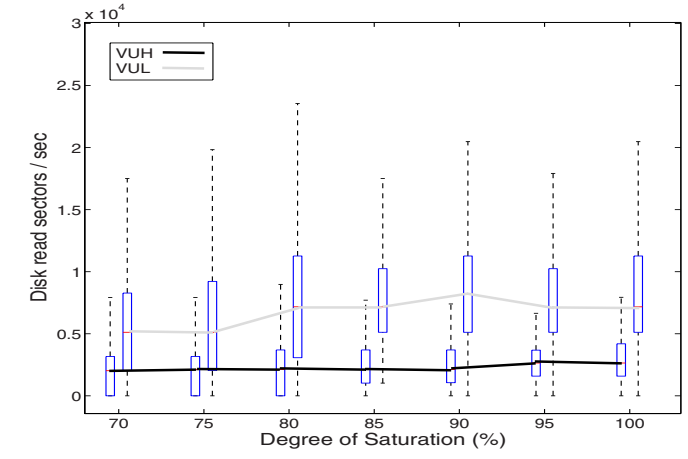
(b) Helix: persistent load spike vs. temporary load spike

Fig. 3. Abnormal behaviors of CPU utilization

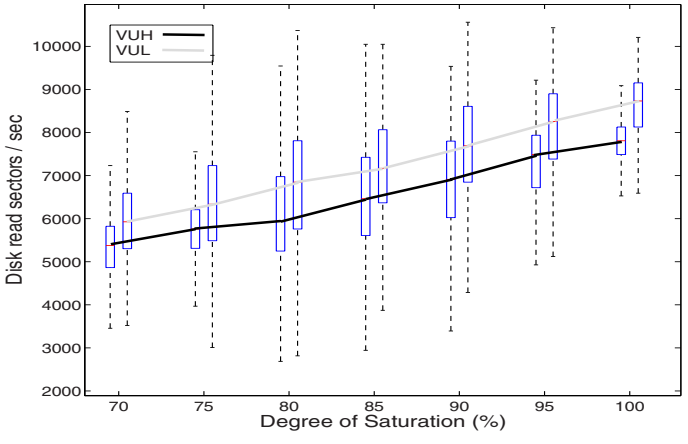
VUH workloads experienced serious data copying overhead, causing a dramatic CPU increase with higher loads and resulting in much lower server saturation compared with the Helix package.

3.3 Server-Side Observations

Server side performance metrics are by far the easiest to identify and understand. CPU, disk, network, and memory identify the critical resources of any modern computer system. Each of those metrics can reach the saturation region independently. CPU utilization indicates whether the server processor can keep up with the tasks associated with serving the streams. Disk and network utilization indicate how much of the available



(a) Darwin



(b) Helix

Fig. 4. Rate of disk sector reads for VUx workloads

bandwidth from these two subcomponents is being used by a particular workload. Memory exhaustion in streaming workloads is an unlikely problem in modern systems. However, if the main memory of the server is exhausted, and the system starts paging, performance deteriorates rapidly and CPU utilization spikes. In fact, memory exhaustion is observed quite often due to the policy of extensive server caching. It is possible to saturate either one of those resources before CPU utilization reaches 100%, and thus they must be monitored independently rather than be proxied by the CPU utilization of the server.

Figure 2 plots the summary statistics of CPU usages for the CPU–intensive pure workloads, depicting different usage patterns of the server processors. They do not show statistically significant trends over the workload ranges of interest. The linear

trend with increasing load on the Darwin server is attributable to the initial load offset, which is incremented as an experiment progresses (Fig. 3(a)). The sawtoothed temporal dependency of load usage makes it hard to estimate the current server state from the performance data collected during a randomly chosen short time interval if the initial load offset is unknown. On the Helix server, the median CPU measurements for LUx workloads shows a slightly negative trend with increasing load (Fig. 2(b)). Furthermore, the CPU usage for the popular workloads (LPx and VPx) is non-monotonic with changing load. The Helix experiments tend to have more load spikes (Fig. 3(b)) than the Darwin experiments. While the temporarily imposed load spike (VUH measurements in the Figure) disappears quickly, a persistent spike lasts for a long time and the system stays heavily loaded, shown as a '+' symbol in the Fig. 2(b). Of course, the non-monotonic nature of the Helix server is the side-effect of such persistent spikes.

The disk-read measurements (Fig. 4) show semi-monotonic trends with increased load. While the Helix measurements show a steady curve, the Darwin measurements report wider variability of the disk read rate. Compared with the network traffic, the disk read traffic is less reliable in providing enough prediction power to estimate the current load. The disk-write measurements are very reliable, consistent, and identifiable for VUx workloads because they reflect how many unique files are opened. In our test environments, all opened inode data structures are written back to the disks periodically, which causes disk-write traffic. Obviously, this server performance metric helps to identify VUx workloads and their load accurately.

Figure 5 shows the change of cache memory size for the VUL workloads. Memory usage behavior uniquely characterizes different memory management policies implemented in each server: the Darwin server uses fixed amounts of cache memory for each session and the Helix server uses all available memory space to maximize the caching effect. We conclude that the memory measurements of the Darwin server estimate the number of VUx workload sessions accurately without any a priori knowledge on the

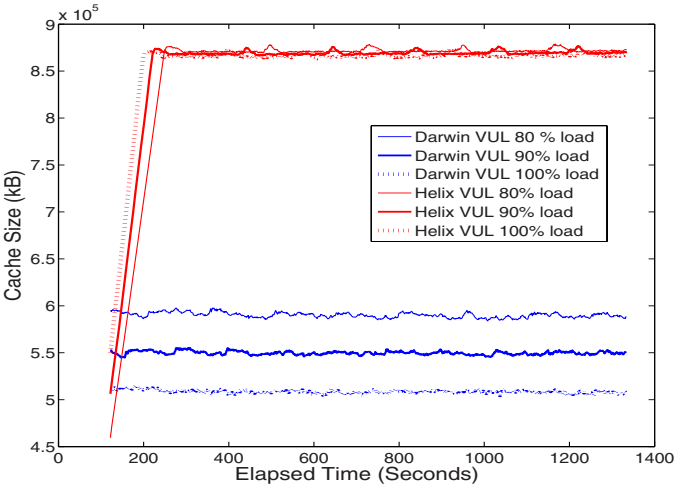
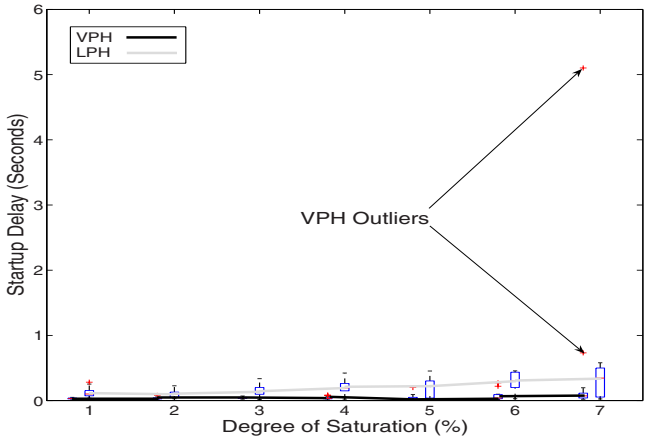
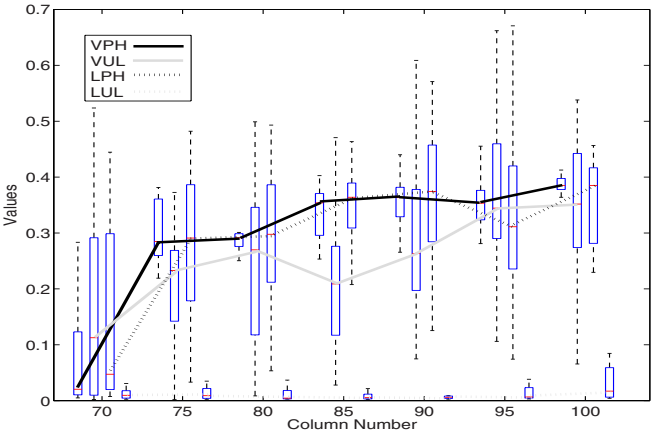


Fig. 5. Different cache memory usage policies observed for VUL workloads



(a) Darwin



(b) Helix

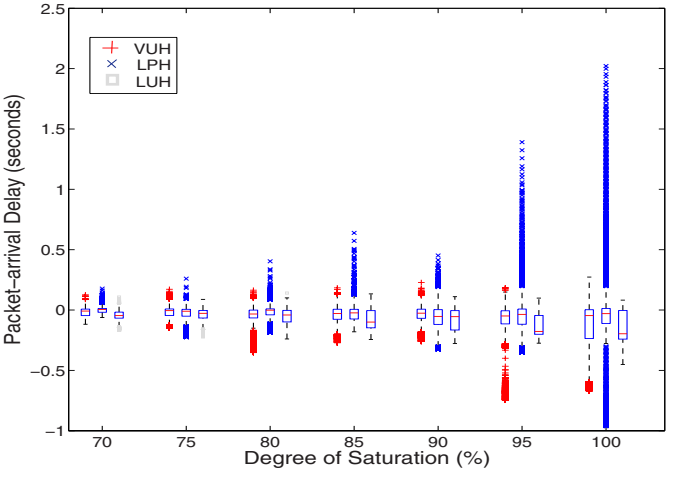
Fig. 6. Startup Delay

imposed workloads. However, those of the Helix server collected in the middle of the experimental run are useless for the server system estimation.

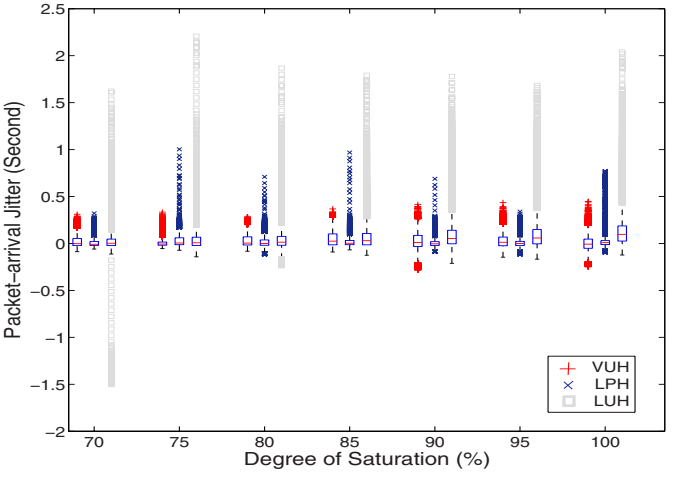
3.4 Client-Side Observations

We expect that there are good indicators at the client for the server saturation. Depending on server management policy, an overload may result in increased startup latency or a number of late packets, or both.

Specifically, we are interested in two client-side performance metrics: startup delay and jitter. Increased startup latency indicates that the server is falling behind in processing new requests. Jitter captures the delay in packets sent by the server as seen by the



(a) Darwin



(b) Helix

Fig. 7. Probe-client packet-arrival jitter

client. For a stream encoded at a particular rate, each packet is expected to be sent by the server and arrive at the client with a predetermined deadline. Late packets indicate that something could be going wrong and may be an early sign of more serious problems.

In Fig. 6(a), the Darwin server shows a number of outliers from the startup delays when approaching the saturated region for VPx workloads. When the Darwin server is fully loaded with the VPH workload, startup delays begin to show extremely large outliers (5 seconds and 0.8 seconds in Fig. 6(a)), while their median values are very small. We cannot observe such large outliers when the server runs in an unsaturated

region. Thus, any occurrence of intolerable startup delay for the VPx workload on the Darwin server would quickly indicate that the server system enters the saturated region. Figure 6(b) shows that the median values of the startup delays over various workloads on the Helix server seem to converge quickly when the load approaches 75% of the saturation level. Thus, any median values (more than 300 milli-seconds) collected for a short-period of time indicate that the server experiences more than 70% of the saturating load. However, the wide variability and the negative trends of the Helix startup delay above the 75% load-percentile inevitably prevent any predictions.

Figure 7 shows the median values and their variabilities of the packet-arrival delays. Jitter corresponding to zero means that a packet arrived exactly at the desired time; a negative value means that it arrived earlier; and a positive value means that it arrived late. The median value of the Darwin server tends to be skewed for the unpopular workloads and its interquantile range (IQR) becomes negatively larger. The Darwin LPH workload shows a semi-monotonic trend of the outliers with increasing load. The Helix experiments tend to have more late outliers than the Darwin experiments, but otherwise, there is no statistical significance of the median, IQR, and the outliers.

4 Prediction Model

In Section 3 we found that individual server or client performance metrics could not estimate the current server state reliably or unambiguously. Therefore, we propose a mathematical model that combines a large number of measurements and from these deduces "virtual resources" with greater predictive power.

First, we annotate the calibrated measurement data collected from both server and clients with a normalized client workload vector. We start with eight nominally distinct saturation resource directions (or eight virtual resources), one for each of the eight pure workloads. At saturation, each pure client workload must use 100% of the resource direction on which it saturates. It can also use between 0% and 100% of the other resource directions at saturation. In all cases, the usage of a virtual resource is separately constrained both to be an affine function of the measurement vector and to be an affine function of the client workload level. Using projection-onto-convex sets, we solve this problem using robust, total least squares under inequality and vector-norm constraints. The inequality constraints on the robust total-least squares include the constraint for non-negative, non-oversaturating resource usage at the same time as finding the measurement-to-resource models. The model for the client-to-resource usage is then refined in the alternate projection step, using the resource usage estimates derived from the measurements and the most recent measurement-to-resource models along with the actual client workloads. After completing this process for a fixed number of resource directions, we consider lowering the number of resource directions by merging directions that are similar. We measure this similarity in direction using the correlation coefficients on the resource usage across all client types. If the correlation in client usage across two resource dimensions is greater than 90%, we merge the two resource dimensions together, and repeat the whole model estimation process with this smaller number of resource dimensions.

The result of this modeling was four measurement-to-resource models and a matrix description of client-resource-usage (see Tab. 3). This model results in much more powerful prediction capabilities. For example, compared with UDP-traffic measurements whose standard deviation (representing a prediction error) is reported as 9% for the LPx workload, the usage-prediction error of our resource model V1 is significantly improved to 0.8%. We validated the usefulness of this model by comparing the saturation decision results from the experiments running a mixture of pure workloads. A more detailed description of this model is contained in [8].

Table 3. Virtual resource usage matrix for the Helix server. The percentage listed here is the amount of virtual resources that each pure client workload uses when loaded to saturation.

	Pure workload							
Resource	VPH	VPL	VUH	VUL	LPH	LPL	LUH	LUL
V1	29	36	0	2	100	100	8	6
V2	28	24	100	100	1	2	71	43
V3	100	100	1	13	98	99	87	51
V4	17	5	39	4	30	50	100	100

5 Related Work

Most video server admission policies, such as the one proposed by Zimmermann *et al.* [9], rely on the deterministic or the statistical modeling of disk drive I/O performance. While these models provide an extremely accurate estimation of disk loads, their simulation or evaluation results are impacted by input parameters and different system components such as the CPU load or memory limits. Our main observations is that there exist many different factors that cause system saturation for heterogeneous workloads and hence evaluating a single resource is insufficient to obtain accurate results. Our model, on the other hand, proposes a manageable number of different pure workloads and then identifies the critical saturation causes for each.

Cherkasova *et al.* [10,11] provided one of the first comprehensive performance analysis of media servers under video-on-demand workloads with both popular and unpopular content. The authors identified important client side performance metrics, namely jitter and rebuffering. The paper also recognized the need to measure the basic capacity of the server under different workloads. Our work extends both the workload space by examining live streams in addition to video on demand (as well as considering their mix), and the client metrics space by looking into failures, startup latency, and thinning. We conjecture that one needs to look at all the important client side metrics since different ones may be affected under different workloads or on different streaming servers.

Independent monitoring and verification of performance is provided by several commercial services such as Keynote, Streamcheck, and Broadstream. Keynote is a leader in the field of world wide monitoring of web and streaming media performance by measuring startup times, audio and video bandwidth, and packet counts (delivered, recovered, and late). They also provide a weighted score that summarizes in a single

number the overall the performance derived from low level metrics. Streamcheck does similar probing for performance while Broadstream focuses on their IP TV network and measures performance at each receiver.

6 Conclusions

In this article we have presented a systematic performance evaluation methodology to measure the capacity of streaming media systems consistently and reliably. We then validated our methodology with a case study of two commercial streaming servers.

Our rigorous methodology consists of three consecutive steps: three dimensional workload selection, a quantifiable, reproducible server capacity decision model, and finally a calibrated data collection. Among them, elaborate server capacity decision model plays a key role in deriving a correct and final capacity limit. Without it, we have witnessed a lot of wrong server capacity decision due to an unexpected temporary performance spike during the experiments even we isolated the experiments.

Compared with our earlier approach that primarily relied on expert's intuition, our new method correctly predicts a 52% higher server capacity for the *Darwin* VUL workload while confirming the other workload decisions. In addition, our new method establishes an efficient evaluation methodology that quantifies a user's streaming experience, mainly for the *Helix* experiments. We have demonstrated that the performance metrics at the server-side such as CPU load , I/O traffic, and memory usage and at the client-side such as startup delay are affected by the system load in different ways, and that each by itself cannot be a good classifier to differentiate workload types and to estimate the system load accurately. Instead, we use a data-mining based virtual resource model.

The lessons we learnt through the extensive measurements of two commercial streaming servers are directly applicable to the management of multiple servers (e.g., in a cluster configuration). We conclude that better throughput can be achieved by assigning requests and content so that the popularity of clips is maximized, by separating requests for on-demand and live streams to different servers and by converting on-demand requests to live streams whenever possible (e.g., by regularly scheduling live streams for very popular content).

In the near future we plan to optimize our server capacity decision model and to build cluster server management policies that optimize performance by carefully assigning stream requests to appropriate servers based on the request characteristics and the current server states.

References

1. Covell, M., Roy, S., Seo, B.: Predictive modeling of streaming servers. SIGMETRICS Performance Evaluation. Review 33(2), 33–35 (2005)
2. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (1996)
3. P.862, I.T.R.: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs (2001)

4. Wolf, S.: Draft technical report for 'objective video quality measurement using a peak-signal-to-noise-ratio (PSNR) full reference technique'. T1A1.1/2001-026R1 (2001)
5. Keynote Inc.: Measurement and Monitoring: Streaming Perspective (2003), http://www.keynote.com/downloads/datasheets/streaming_0104.pdf
6. Apple: Darwin Streaming Server 4.1.3 (2003), http://developer.apple.com/darwin/projects/streaming/
7. RealNetworks: Helix Universal Server, http://www.realnetworks.com/products/media_delivery.html
8. Covell, M., Seo, B., Roy, S., Spasojevic, M., Kontothanassis, L., Bhatti, N., Zimmermann, R.: Calibration and prediction of streaming-server performance. Technical report, HP Laboratories Palo Alto (2005)
9. Zimmermann, R., Fu, K.: Comprehensive statistical admission control for streaming media servers. In: Proceedings of the ACM Multimedia 2003 (2003)
10. Cherkasova, L., Staley, L.: Building a performance model of streaming media application in utility data center environment. In: 3rd IEEE/ACM International Symp. on Cluster Computing and the Grid, Tokyo, Japan (2003)
11. Cherkasova, L., Tang, W., Vahdat, A.: Mediaguard: a model–based framework for building streaming media services. In: Proc. of the SPIE Conf. on Multi–Media Computing and Networking (2005)

Part II

Artificial Intelligence and Decision Support Systems

On Processing Temporal Observations in Monitoring of Discrete-Event Systems

Gianfranco Lamperti and Marina Zanella

Dipartimento di Elettronica per l'Automazione, Via Branze 38, 25123 Brescia, Italy
{lamperti,zanella}@ing.unibs.it

Abstract. Observations play a major role in monitoring and diagnosis of discrete-event systems (DESs). In a distributed, large-scale setting, the observation of a DES over a time interval is not perceived as a totally-ordered sequence of observable labels but, rather, as a directed acyclic graph, under uncertainty conditions. Problem solving, however, requires generating a surrogate of such a graph, the index space. Furthermore, the observation hypothesized so far has to be integrated at the reception of a new fragment of observation. This translates to the need for computing a new index space every time. Since such a computation is expensive, a naive generation of the index space from scratch at the occurrence of each observation fragment becomes prohibitive in real applications. To cope with this problem, the paper introduces an incremental technique for efficiently modeling and indexing temporal observations of DESs.

1 Introduction

Observations are the inputs to several tasks that can be carried out by exploiting Model-Based Reasoning techniques. Typically, they are the inputs to supervision, control and monitoring of physical processes [1]; they represent the symptoms of diagnosis [2]; they are the clues for history reconstruction [3,4], and the test cases for software debugging [5,6]. *Temporal observations* refer to dynamical systems and processes, and are endowed not only with a logical content, describing *what* has been observed, but also with a temporal content, describing *when* it has been observed. Both (independent) aspects, can be modeled either quantitatively or qualitatively. This paper addresses the most qualitative abstraction of the notion of a temporal observation, i.e. an observation whose logical and temporal contents are both qualitative. This abstraction is quite important since adopting qualitative models is an issue of Model-Based Reasoning and Qualitative Physics as well, in far as reasoning about (a finite number of) qualitative values is easier and computationally cheaper. Moreover, adopting a higher abstraction level first, so as to focus attention, and a more detailed level later, is a principle of hierarchical model-based diagnosis [7]. A general model for (qualitative uncertain) temporal observations was proposed in [8], and exploited for describing the input of an a-posteriori diagnosis task. Such a model consists of a directed acyclic graph where each node contains an uncertain logical content and each edge is a temporal precedence relationship. Thus, the graph, altogether, shows all the uncertain values observed over a time interval and their partial temporal ordering. Each uncertain logical content

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 135–146, 2008.

ranges over a set of qualitative values (labels). Therefore the observation graph implicitly represents all the possible sequences of labels consistent with the received temporal observation, where each sequence is a sentence of a language. Then, the observation graph, although intuitive and easy to build from the point of view of the observer, is unsuitable for processing. For any further processing it is better to represent a language in the standard way regular languages are represented [9], that is, by means of a deterministic automaton. In [8] this automaton is called *index space* and it is built as the transformation of a nondeterministic automaton drawn from the observation graph. The problem arises when the nodes of the observation graph are received one at a time, typically in monitoring and diagnosis of dynamical systems, in particular, discrete-event systems (DESs). In fact, the supervision process is required to react at each occurring piece of observation so as to generate appropriate diagnostic information [10,11]. This translates to the need for generating a new index space at each new reception. However, a naive approach, that each time makes up the new index space from scratch, would be inadequate from the computational point of view. We need therefore an incremental technique for index-space generation.

2 Application Domain

A reference application domain of power networks is considered. A power network is composed of transmission lines. Each line is protected by two breakers that are commanded by a protection. The protection is designed to detect the occurrence of a short circuit on the line based on the continuous measurement of its impedance: when the impedance goes beyond a given threshold, the two breakers are commanded to open, thereby causing the extinction of the short circuit. In a simplified view, the network is represented by a series of lines, each one associated with a protection, as displayed in Fig. 1, where lines $l_1 \cdots l_4$ are protected by protections $p_1 \cdots p_4$, respectively. For instance, p_2 controls l_2 by operating breakers b_{21} and b_{22}. In normal (correct) behavior, both breakers are expected to open when tripped by the protection. However, the protection system may exhibit an abnormal (faulty) behavior, for example, one breaker or both may not open when required. In such a case, each faulty breaker informs the protection about its own misbehavior. Then, the protection sends a request of recovery actions to the neighboring protections, which will operate their own breakers appropriately. For example, if p_2 operates b_{21} and b_{22} and the latter is faulty, then p_2 will send a signal to p_3, which is supposed to command b_{32}. A recovery action may be faulty on its turn. For example, b_{32} may not open when tripped by p_3, thereby causing a further propagation of the recovery to protection p_4. The protection system is designed to propagate the recovery request until the tripped breaker opens correctly. When the protection system is reacting, a subset of the occurring events are visible to the operator in a control room who is in charge of monitoring the behavior of the network and, possibly, to issue explicit commands so as to minimize the extent of the isolated subnetwork. Typical visible events are *short* (a short circuit occurred on the line), *open* (a breaker opened), *close* (a breaker closed), and *end* (the short circuit extinguished). Generally speaking, however, the localization of the short circuit and the identification of the faulty breakers may be impractical in real contexts, especially when the extent

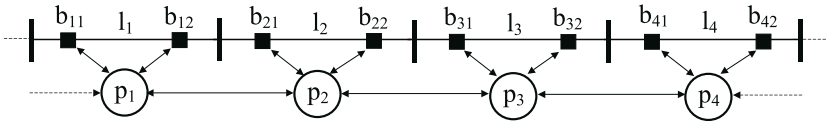


Fig. 1. Power transmission network

of the isolation spans several lines and the operator is required to take recovery actions within stringent time constraints. On the one hand, there is the problem of observability: the observable events generated during the reaction of the protection system are generally incomplete and uncertain in nature. On the other, whatever the observation, it is impractical for the operator to reason on the observation itself so as to make consistent hypotheses on the behavior of the system and, eventually, to establish the shorted line and the faulty breakers.

3 Temporal Observations

A temporal observation $\mathcal{O}$ is the mode in which the observable labels, generated by the evolution of a DES, are perceived by the observer. Considering the realm of asynchronous DESs, such as *active systems* [12], a *history* $h = \langle T_1, \ldots, T_m \rangle$ of a system is a sequence of component transitions, where each transition refers to a communicating automaton. Since a subset of the transitions are visible, the system history is expected to generate a sequence of observable labels, namely a *temporal sequence* $\langle \ell_1, \ldots, \ell_k \rangle$, $k \leq m$, where each ℓ_i, $i \in [1\,..\,k]$, is the label generated by a visible transition in h. However, due to the multiplicity of communication channels between the (distributed) system and the observer, and to noise on such channels, the temporal observation $\mathcal{O}$ received by the observer is likely to differ from the temporal sequence generated by the system. Intuitively, $\mathcal{O}$ is a sequence of temporal fragments bringing information about what/when something is observed. Formally, let Λ be a domain of observable labels, including the *null label* ε, which is invisible to the observer. A *temporal fragment* φ is a pair (λ, τ), where $\lambda \subseteq \Lambda$ is called the *logical content*, and τ is a set of fragments, called the *temporal content*. Specifically, $\mathcal{O}$ is a sequence of temporal fragments, $\mathcal{O} = \langle \varphi_1, \ldots, \varphi_n \rangle$, such that $\forall i \in [1\,..\,n], \varphi_i = (\lambda_i, \tau_i)\ \ (\tau_i \subseteq \{\varphi_1, \ldots, \varphi_{i-1}\})$. The temporal content of a fragment φ is supposed to refer to a (possibly empty) subset of the fragments preceding φ in $\mathcal{O}$. Thus, a fragment is uncertain in nature, both logically and temporally. Logical uncertainty means that λ includes the actual (possibly null) label generated by a system transition, but further spurious labels may be involved too. Temporal uncertainty means that the actual label was generated by the system transition after all the actual labels relevant to the fragments in τ had been generated. As such, both logical and temporal uncertainty are a sort of relaxation of the temporal sequence generated by the system, the former relaxing the actual visible label into a set of candidate labels, the latter relaxing absolute temporal ordering into partial ordering.

A *sub-observation* $\mathcal{O}_{[i]}$ of $\mathcal{O}$, $i \in [0\,..\,n]$, is the (possibly empty) prefix of $\mathcal{O}$ up to the i-th fragment, $\mathcal{O}_{[i]} = \langle \varphi_1, \ldots, \varphi_i \rangle$.

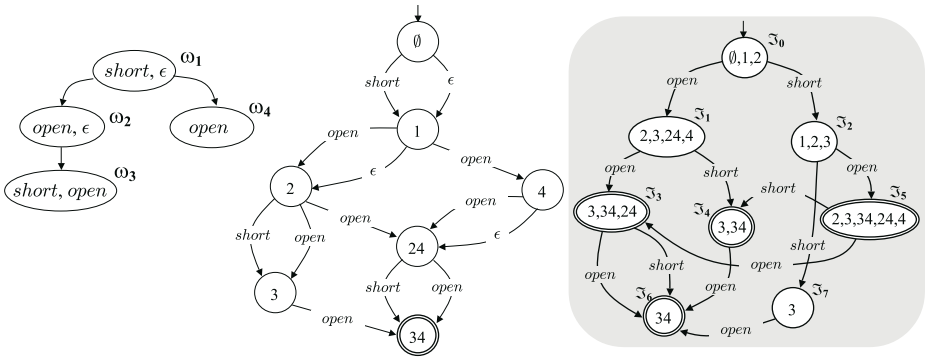


Fig. 2. From left to right, $\gamma(\mathcal{O})$, $Psp(\mathcal{O})$, and $Isp(\mathcal{O})$

Example 1. Let $\Lambda = \{\ldots, \ldots, \varepsilon\}$, $\mathcal{O} = \langle\varphi_1, \varphi_2, \varphi_3, \varphi_4\rangle$, where $\varphi_1 = (\{\ldots, \varepsilon\}, \emptyset)$, $\varphi_2 = (\{\ldots, \varepsilon\}, \{\varphi_1\})$, $\varphi_3 = (\{\ldots, \ldots\}, \{\varphi_2\})$, $\varphi_4 = (\{\ldots\}, \{\varphi_1\})$. φ_1, is logically uncertain (either *short* or nothing has been generated). φ_2 follows φ_1 and is logically uncertain (vs. nothing). φ_3 follows φ_2 and is logically uncertain (*short* vs. *open*). φ_4 follows φ_1 and is logically certain (*open*). However, no temporal relationship is defined between φ_4 and φ_2 or φ_3.

Based on Λ, a temporal observation $\mathcal{O} = \langle\varphi_1, \ldots, \varphi_n\rangle$ can be represented by a DAG, called an *observation graph*,

$$\gamma(\mathcal{O}) = (\Lambda, \Omega, \mathbb{E}) \tag{1}$$

where $\Omega = \{\omega_1, \ldots, \omega_n\}$ is the set of nodes isomorphic to the fragments in $\mathcal{O}$, with each node being marked by a nonempty subset of Λ, and $\mathbb{E}$ is the set of edges isomorphic to the temporal content of fragments in $\mathcal{O}$. A *precedence relationship* is defined between nodes of the graph, specifically, $\omega \prec \omega'$ means that $\gamma(\mathcal{O})$ includes a path from ω to ω', while $\omega \preceq \omega'$ means either $\omega \prec \omega'$ or $\omega = \omega'$.

Example 2. Consider the observation $\mathcal{O}$ defined in Example 1. The relevant observation graph $\gamma(\mathcal{O})$ is shown on the left of Fig. 2. Note how $\gamma(\mathcal{O})$ implicitly contains several candidate temporal sequences, with each candidate sequence being generated by picking up a label from each node of the graph without violating the partially-ordered temporal relationships among nodes. Possible candidates are, among others, $\langle\ldots, \ldots, \ldots, \ldots\rangle$, $\langle\ldots, \ldots, \ldots\rangle$, and $\langle\ldots, \ldots\rangle$.[1] However, we do not know which of the candidates is the actual temporal sequence generated by the system, with the other ones being the *spurious* candidate sequences. Consequently, from the observer viewpoint, all candidate sequences share the same ontological status.

[1] The fact that the length of a candidate temporal sequence may be shorter than the number of nodes in the observation graph comes from the immateriality of the null label ε, which is 'transparent'. For instance, candidate $\langle\varepsilon, \varepsilon, short, open\rangle$ is in fact $\langle short, open\rangle$.

4 Indexing Observations

The rationale of the paper is that, both for computational and space reasons, *the observation graph is inconvenient for carrying out a task that takes as input a temporal observation*. This claim applies to *linear observations* as well, which are merely a sequence O of observable labels. In this case, it is more appropriate to represent each sub-observation $O' \subseteq O$ as an integer index i corresponding to the length of O'. As such, i is a surrogate of O'. The same approach has been proposed for graph-based temporal observations [13]. However, we need extending the notion of an index appropriately and make model-based reasoning on a surrogate of the temporal observation, called an index space.

Let $\gamma(\mathcal{O}) = (\Lambda, \Omega, \mathbb{E})$ be an observation graph. A *prefix* $\mathcal{P}$ of $\mathcal{O}$ is a (possibly empty) subset of Ω where $\forall\, \omega \in \mathcal{P}\, (\nexists \omega' \in \mathcal{P}\ (\omega' \prec \omega))$. The formal definition of an index space is supported by two functions on $\mathcal{P}$. The set of *consumed nodes* up to $\mathcal{P}$ is

$$, \ldots (\mathcal{P}) = \{\omega \mid \omega \in \Omega, \omega' \in \mathcal{P}, \omega \preceq \omega'\}. \tag{2}$$

The set of *consumable nodes* from $\mathcal{P}$, called the *frontier* of $\mathcal{P}$, is defined as

$$\ldots \cdot (\mathcal{P}) = \{\omega \mid \omega \in (\Omega -, \ldots (\mathcal{P})), \forall (\omega' \mapsto \omega) \in \mathbb{E}\ (\omega' \in, \ldots (\mathcal{P}))\}. \tag{3}$$

Example 3. Considering $\gamma(\mathcal{O})$ in Fig. 2, with $\mathcal{P} = \{\omega_2, \omega_4\}$, we have $, \ldots (\mathcal{P}) = \{\omega_1, \omega_2, \omega_4\}$ and $\ldots \cdot (\mathcal{P}) = \{\omega_3\}$.

Let $\mathcal{O}$ be a temporal observation. The *prefix space* of $\mathcal{O}$ is the nondeterministic automaton

$$, \, , (\mathcal{O}) = (\mathbb{S}^{\mathrm{n}}, \mathbb{L}^{\mathrm{n}}, \mathbb{T}^{\mathrm{n}}, S_0^{\mathrm{n}}, S_{\mathrm{f}}^{\mathrm{n}}) \tag{4}$$

where $\mathbb{S}^{\mathrm{n}} = \{\mathcal{P} \mid \mathcal{P} \text{ is a prefix of } \mathcal{O}\}$ is the set of states, $\mathbb{L}^{\mathrm{n}} = \{\ell \mid \ell \in \lambda, (\lambda, \tau) \in \Omega\}$ is the set of labels, $S_0^{\mathrm{n}} = \emptyset$ is the initial state, $S_{\mathrm{f}}^{\mathrm{n}} = \mathcal{P} \in \mathbb{S}^{\mathrm{n}}$ such that $, \ldots (\mathcal{P}) = \Omega$, is the final state, and $\mathbb{T}^{\mathrm{n}} : \mathbb{S}^{\mathrm{n}} \times \mathbb{L}^{\mathrm{n}} \mapsto 2^{\mathbb{S}^{\mathrm{n}}}$ is the transition function such that $\mathcal{P} \xrightarrow{\ell} \mathcal{P}' \in \mathbb{T}^{\mathrm{n}}$ iff, defining the '$\oplus$' operation as

$$\mathcal{P} \oplus \omega = (\mathcal{P} \cup \{\omega\}) - \{\omega' \mid \omega' \in \mathcal{P}, \omega' \prec \omega\} \tag{5}$$

we have $\omega \in \ldots \cdot (\mathcal{P}), \omega = (\lambda, \tau), \ell \in \lambda, \mathcal{P}' = \mathcal{P} \oplus \omega$. The *index space* of $\mathcal{O}$ is the deterministic automaton

$$, \, (\mathcal{O}) = (\mathbb{S}, \mathbb{L}, \mathbb{T}, S_0, \mathbb{S}_{\mathrm{f}}) \tag{6}$$

equivalent to $, \, , (\mathcal{O})$. Each state in $, \, (\mathcal{O})$ is an index of $\mathcal{O}$. The peculiarity of an index space is that each path from S_0 to a final state is a mode in which we may choose a label in each node of the observation graph $\gamma(\mathcal{O})$ based on the partial ordering imposed by $\gamma(\mathcal{O})$ [8].

Example 4. Consider $\gamma(\mathcal{O})$ in Fig. 2. Shown on the same figure are the prefix space $, \, , (\mathcal{O})$ (center) and the index space $, \, (\mathcal{O})$ (right). Each prefix is written as a string of digits, e.g., 24 stands for $\mathcal{P} = \{\omega_2, \omega_4\}$. Final states are double circled. According to the standard algorithm that transforms a nondeterministic automaton into a deterministic one [9], each node of $, \, (\mathcal{O})$ is identified by a subset of the nodes of $, \, , (\mathcal{O})$. Nodes in $, \, (\mathcal{O})$ have been named $\Im_0 \cdots \Im_7$. These are the indexes of $\mathcal{O}$.

5 Incremental Indexing

In case we need to compute the index space of each sub-observation of $\mathcal{O} = \langle \varphi_1, \ldots, \varphi_n \rangle$, namely $\mathcal{I}(\mathcal{O}_{[i]})$, $i \in [1\,..\,n]$, the point is, *it is prohibitive to calculate each new index space from scratch at the occurrence of each fragment φ_i, as this implies the construction of the nondeterministic $\mathcal{I}(\mathcal{O}_{[i]})$ and its transformation into the deterministic $\mathcal{I}(\mathcal{O}_{[i]})$*. A better approach is generating the new index space incrementally, based on the previous index space and the new observation fragment, avoiding the generation and transformation of the nondeterministic automaton. This is performed by algorithm *Increment*, generating the new observation graph $\gamma(\mathcal{O}_{[i]})$ and relevant index space $\mathcal{I}(\mathcal{O}_{[i]})$, based on the previous $\gamma(\mathcal{O}_{[i-1]})$ and $\mathcal{I}(\mathcal{O}_{[i-1]})$, and the new fragment φ_i, as specified below.

$ncrement(\gamma(\mathcal{O}_{[i-1]}), \mathcal{I}(\mathcal{O}_{[i-1]}), \varphi_i) \Rightarrow (\gamma(\mathcal{O}_{[i]}), \mathcal{I}(\mathcal{O}_{[i]}))$
begin
Yield $\gamma(\mathcal{O}_{[i]})$ based on the new fragment $\varphi_i = (\lambda_i, \tau_i)$;
Let ω_i be the node of $\gamma(\mathcal{O}_{[i]})$ corresponding to φ_i;
Initialize $\mathcal{I}(\mathcal{O}_{[i]})$ as a copy of $\mathcal{I}(\mathcal{O}_{[i-1]})$;
$\mathbb{B} := \{(N, \mathcal{P}, \omega_i) \mid N \in \mathcal{I}(\mathcal{O}_{[i]}), \mathcal{P} \in N, \omega_i \in \ldots(\mathcal{P})\}$;
loop
Pick up a bud $B = (N, \mathcal{P}, \omega)$, $\omega = (\lambda, \tau)$, from $\mathbb{B}$;
$\mathcal{P}' := \mathcal{P} \oplus \omega$;
for each $\ell \in \lambda$ **do**
Extend $\mathcal{I}(\mathcal{O}_{[i]})$ based on rules $(\mathcal{R}_1)\,..\,(\mathcal{R}_9)$ below:
$(\mathcal{R}_1)$ **if** $\ell = \varepsilon, \exists N' = N \cup \{\mathcal{P}'\}, N' \neq N$ **then**
Merge N and N'; Update $\mathbb{B}$
$(\mathcal{R}_2)$ **elsif** $\ell = \varepsilon, \nexists N' = N \cup \{\mathcal{P}'\}$ **then**
Extend N with $\mathcal{P}'$; Update $\mathbb{B}$
$(\mathcal{R}_3)$ **elsif** $\ell \neq \varepsilon, \nexists$ edge leaving N marked by $\ell, \exists N' = \{\mathcal{P}'\}$ **then**
Create a new edge $N \xrightarrow{\ell} \{\mathcal{P}'\}$
$(\mathcal{R}_4)$ **elsif** $\ell \neq \varepsilon, \nexists$ edge leaving N marked by $\ell, \nexists N' = \{\mathcal{P}'\}$ **then**
Create $N' = \{\mathcal{P}'\}$; Create $N \xrightarrow{\ell} N'$; Update $\mathbb{B}$
$(\mathcal{R}_5)$ **elsif** $\ell \neq \varepsilon, \exists N \xrightarrow{\ell} N', \nexists$ another edge to N', $\exists \bar{N} = N' \cup \{\mathcal{P}'\}, \bar{N} \neq N'$ **then**
Merge N' and $\bar{N}$; Update $\mathbb{B}$
$(\mathcal{R}_6)$ **elsif** $\ell \neq \varepsilon, \exists N \xrightarrow{\ell} N', \nexists$ another edge to $N', \nexists \bar{N} = N' \cup \{\mathcal{P}'\}$ **then**
Insert $\mathcal{P}'$ into N'; Update $\mathbb{B}$
$(\mathcal{R}_7)$ **elsif** $\ell \neq \varepsilon, \exists N \xrightarrow{\ell} N', \exists$ another edge to N', $\exists \bar{N} = N' \cup \{\mathcal{P}'\}, \bar{N} \neq N'$ **then**
Replace $N \xrightarrow{\ell} N'$ with $N \xrightarrow{\ell} \bar{N}$
$(\mathcal{R}_8)$ **elsif** $\ell \neq \varepsilon, \exists N \xrightarrow{\ell} N', \exists$ another edge to $N', \exists N'' = N' \cup \{\mathcal{P}'\}$ **then**
Redirect $N \xrightarrow{\ell} N'$ towards N''

($\mathcal{R}_9$) **elsif** $\ell \neq \varepsilon$, $\exists N \xrightarrow{\ell} N'$, $\exists$ another edge to N', $\nexists N'' = N' \cup \{\mathcal{P}'\}$ **then**
Create $N'' = N' \cup \{\mathcal{P}'\}$; Redirect $N \xrightarrow{\ell} N'$ towards N'';
Duplicate for N'' each transition exiting N'; Update $\mathbb{B}$
end-if
end for;
Remove bud B from $\mathbb{B}$
while $\mathbb{B} \neq \emptyset$;
Yield the final states of $\gamma(\mathcal{O}_{[i]})$
end

Increment is supported by a data structure, the *bud set*, denoted $\mathbb{B}$. Each bud in $\mathbb{B}$ is a triple $(N, \mathcal{P}, \omega)$, where N is a node of the index space, $\mathcal{P}$ a prefix in N, and ω a node of the observation graph belonging to the frontier of $\mathcal{P}$. Note that initially all the buds share the same ω_i, which is the node of the observation graph corresponding to the new fragment φ_i. Each of such buds indicates that, owing to the new observation fragment, N needs further processing. This means, for instance, that all the candidate sequences of labels up to N are followed by a label belonging to the logical content of the new fragment. Therefore, N has to be extended, possibly by new edges, leading to old/new nodes. Once processed, the bud is removed from $\mathbb{B}$. However, processing a bud possibly causes the generation of new buds since, for instance, the candidate sequences of labels up to a newly created node N', ending with a label of the new fragment, can be followed in the emission order by labels inherent to fragments received before it. Therefore, also N' has to be extended. Each rule $\mathcal{R}_i$, $i \in [1\,..\,9]$, is an association *condition-action*. The conditions are mutually exclusive. They involve the current topology of the index space, the bud $B = (N, \mathcal{P}, \omega)$ picked up at the beginning of the body of the main loop, the new prefix $\mathcal{P}'$, and label $\ell \in \lambda$, being $\omega = (\lambda, \tau)$. If no condition holds, then no operation is performed. For instance, the action of $\mathcal{R}_1$ merges nodes N and N'. To do so, all edges entering/leaving N are redirected to/from N', while N is removed. This may cause nondeterminism in the transitions exiting N', which is to be removed. After the merging, the bud set must be updated (all the buds inherent to the removed node are removed as well). The action of $\mathcal{R}_9$, instead, redirects the edge $N \xrightarrow{\ell} N'$ towards the new node $N' \cup \{\mathcal{P}'\}$, and duplicates the edges leaving N'. This operation too requires updating the bud set. When the loop terminates, the new index space $\gamma(\mathcal{O}_{[i]})$ is topologically complete. The final states are those containing a prefix $\mathcal{P}$ such that $\delta(\mathcal{P}) = \emptyset$. (However, yielding the final states is irrelevant from the point of view of the next invocation of *Increment*.)

Example 5. Suppose that the sub-observation $\mathcal{O}_{[3]}$ of observation $\mathcal{O}$ of Example 1 has already been received by the observer, one fragment at a time, and that *Increment* has correctly generated $\gamma(\mathcal{O}_{[3]})$ and $\gamma(\mathcal{O}_{[3]})$. Now the fourth (and last) fragment of $\mathcal{O}$, φ_4, is received and *Increment* has to generate $\gamma(\mathcal{O}_{[4]})$. Shaded on the top-left of Fig. 3 is $\gamma(\mathcal{O}_{[4]})$ at the beginning of the main loop, which equals $\gamma(\mathcal{O}_{[3]})$, with some extra information inherent to the bud set $\mathbb{B}$ drawn by processing φ_4. For space reasons, each set of prefixes $\{\mathcal{P}_i, \ldots, \mathcal{P}_k\}$ is represented as the list of digits $i, \ldots, k$. For example, 1, 2, 5 stands for $\{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_5\}$. Each bud $(N, \mathcal{P}_j, \omega_i) \in \mathbb{B}$ is represented by j^i in node N.

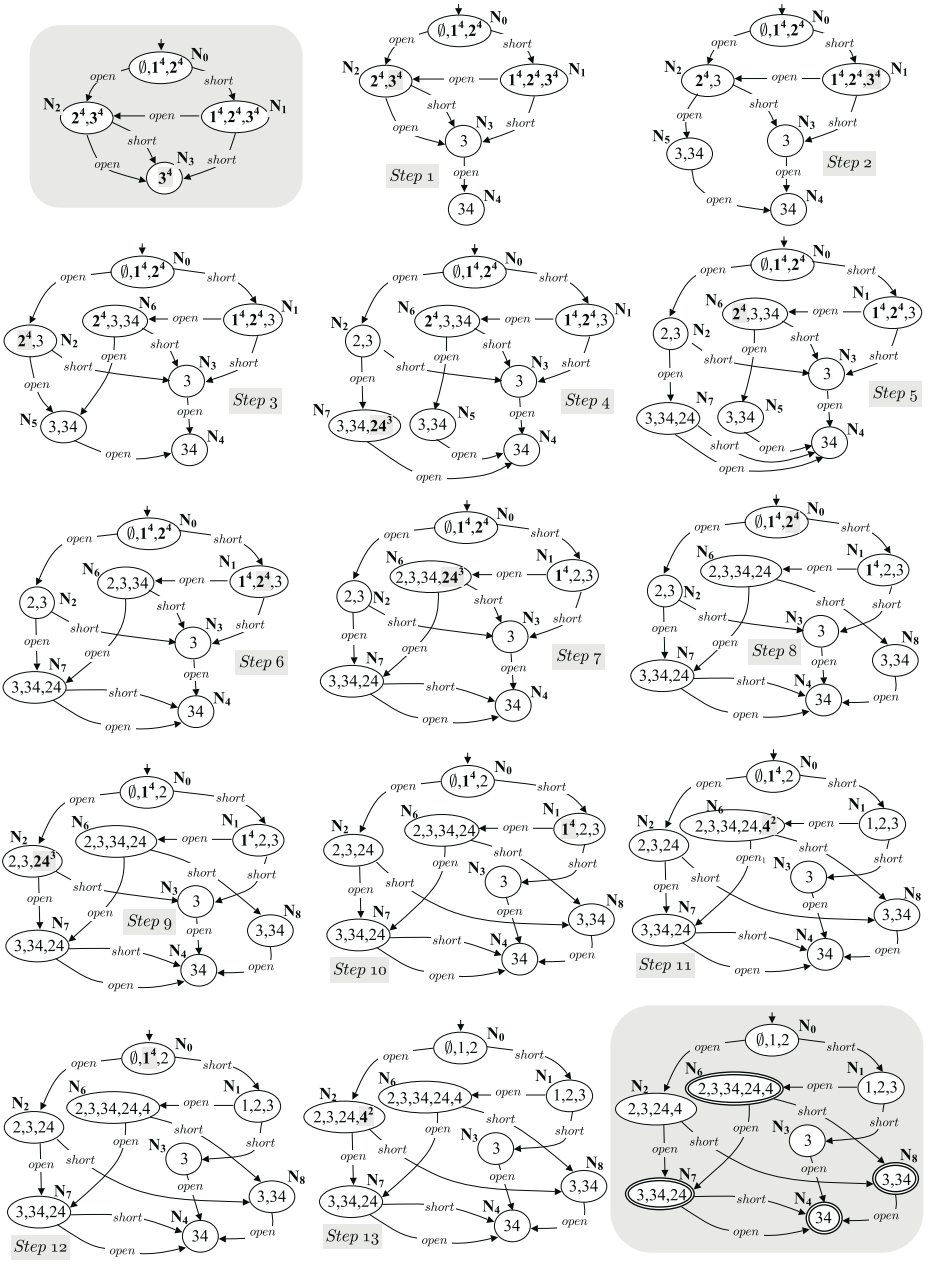


Fig. 3. Tracing of the incremental computation of $Isp(\mathcal{O}_{[4]})$ by the *Increment* algorithm

For example, bud $(N_2, \mathcal{P}_5, \omega_4)$ is written in N_2 as 5^4. The subsequent graphs in Fig. 3 depict the computational state of , $(\mathcal{O}_{[4]})$ at each new iteration of the loop. According to the initial (shaded) graph, at first, $\mathbb{B}$ includes eight buds. The bud B chosen at each

iteration of the main loop is shaded in the corresponding pictorial representation. The loop is iterated fourteen times, as detailed below.

(1) The bud at the first iteration is $(N_3, \mathcal{P}_5, \omega_4)$. $\mathcal{P}' = \mathcal{P}_5 \oplus \omega_4 = \mathcal{P}_6$. Since $\lambda(\omega_4) = \{open\}$, the inner loop is iterated only once, for $\ell =$. This corresponds to rule $\mathcal{R}_4$: the new node N_4 is created and linked from N_3 by an edge marked by , as shown in graph 1 (no new bud is created).
(2) $B = (N_2, \mathcal{P}_5, \omega_4)$, $\lambda = \{$ $\}$, and $\mathcal{P}' = \mathcal{P}_6$. This corresponds to rule $\mathcal{R}_8$: node N_5 is generated (no new bud is created).
(3) $B = (N_1, \mathcal{P}_5, \omega_4)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_6$, rule $\mathcal{R}_8$: N_6 is generated; besides, a new bud $(N_6, \mathcal{P}_2, \omega_4)$ is inserted into $\mathbb{B}$.
(4) $B = (N_2, \mathcal{P}_2, \omega_4)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_3$, rule $\mathcal{R}_8$: node N_7 is generated; moreover, a new bud $(N_7, \mathcal{P}_3, \omega_3)$ is created.
(5) $B = (N_7, \mathcal{P}_3, \omega_3)$, $\lambda = \{$ $\}$, and $\mathcal{P}' = \mathcal{P}_6$. For $\ell =$, this corresponds to rule $\mathcal{R}_3$: edge $N_7 \xrightarrow{open} N_4$ is created. For $\ell =$, no operation.
(6) $B = (N_6, \mathcal{P}_2, \omega_4)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_3$, rule $\mathcal{R}_5$: nodes N_5 and N_7 are merged.
(7) $B = (N_1, \mathcal{P}_2, \omega_4)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_3$, rule $\mathcal{R}_6$: node N_6 is extended with index $\mathcal{P}'$, and a new bud $(N_6, \mathcal{P}_3, \omega_3)$ is created.
(8) $B = (N_6, \mathcal{P}_3, \omega_3)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_6$. For $\ell =$, rule $\mathcal{R}_8$: node N_8 is generated. For $\ell =$, no operation.
(9) $B = (N_0, \mathcal{P}_2, \omega_4)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_3$, and rule $\mathcal{R}_6$: node N_2 is extended with index $\mathcal{P}'$, and a new bud $(N_2, \mathcal{P}_3, \omega_3)$ is created.
(10) $B = (N_2, \mathcal{P}_3, \omega_3)$, $\lambda = \{$ $\}$, and $\mathcal{P}' = \mathcal{P}_6$. For $\ell =$, rule $\mathcal{R}_7$: edge $N_2 \xrightarrow{short} N_3$ is redirected toward N_8. For $\ell =$, no operation.
(11) $B = (N_1, \mathcal{P}_1, \omega_4)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_4$, rule $\mathcal{R}_6$: node N_6 is extended with $\mathcal{P}'$ and bud $(N_6, \mathcal{P}_4, \omega_2)$ is created.
(12) $B = (N_6, \mathcal{P}_4, \omega_2)$, $\lambda = \{$ $, \varepsilon\}$, $\mathcal{P}' = \mathcal{P}_3$. For $\ell =$, no operation. For $\ell = \varepsilon$, no operation.
(13) $B = (N_0, \mathcal{P}_1, \omega_4)$, $\lambda = \{$ $\}$, $\mathcal{P}' = \mathcal{P}_4$, rule $\mathcal{R}_6$: node N_2 is extended with index $\mathcal{P}'$, and a new bud $(N_2, \mathcal{P}_4, \omega_2)$ is created.
(14) $B = (N_2, \mathcal{P}_4, \omega_2)$, $\lambda = \{$ $, \varepsilon\}$, $\mathcal{P}' = \mathcal{P}_3$. For $\ell =$, no operation. For $\ell = \varepsilon$, no operation.

Since $\mathbb{B}$ is empty, the loop terminates. The final states are N_4, N_6, N_7, and N_8.

6 Discussion

The technique for incremental construction of the index space is conceived in the context of dynamic model-based diagnosis of DESs. In this realm, the evolution of a system is monitored based on its model and the observation it generates during operation. The diagnostic engine is expected to react to each new fragment of observation by generating a corresponding set of *candidate diagnoses* based on the previous behavior of the system and the new fragment. As such, the diagnostic process is incremental in nature. Model-based diagnosis of DESs is grounded on two essential elements: the observation $\mathcal{O}$ and the model $\mathfrak{M}$ of the system. Roughly, the diagnostic engine aims to explain $\mathcal{O}$

based on $\mathfrak{M}$. In so doing, a subset of the behavior space of the system is determined and represented by a finite automaton, where each path from the root to a final state is a candidate history of the system. Each history is a sequence of component transitions, where each transition can be either normal or faulty. Consequently, each history corresponds to a candidate diagnosis, namely the set of faulty transitions within the history. Considering a diagnostic problem $\wp(\mathcal{O}, \mathfrak{M})$, where $\mathcal{O} = \langle \varphi_1, \ldots, \varphi_n \rangle$ is a temporal observation, we define the *static solution* of the problem, $\Delta(\wp(\mathcal{O}, \mathfrak{M}))$, the set of candidate diagnoses relevant to the histories drawn from $\mathcal{O}$ based on $\mathfrak{M}$. Since we are interested in updating the set of candidate diagnoses at each new fragment of observation, we have to consider the sequence of static solutions relevant to each *sub-problem* $\wp(\mathcal{O}_{[i]}, \mathfrak{M})$, $i \in [0\,..\,n]$, where $\mathcal{O}_{[i]}$ is the *sub-observation* $\langle \varphi_1, \ldots, \varphi_i \rangle$ up to the i-th fragment. Note that, when $i = 0$, we have an *empty* observation. In other words, the diagnostic engine is expected to generate a new static solution at the occurrence of each newly-generated fragment of observation. This is called the *dynamic solution* of $\wp(\mathcal{O}, \mathfrak{M})$, namely

$$\mathbf{\Delta} = \langle \Delta(\wp(\mathcal{O}_{[0]}, \mathfrak{M})), \ldots, \Delta(\wp(\mathcal{O}_{[n]}, \mathfrak{M})) \rangle. \tag{7}$$

On the other hand, just as the observation graph is not suitable for the diagnostic engine as is, and a surrogate of it (the index space) is used instead, the compositional model $\mathfrak{M}$ turns to be inadequate as is, and a surrogate of it is considered, namely the *model space* of $\mathfrak{M}$, denoted $\mathcal{M}(\mathfrak{M})$. Thus, for computational reasons, the diagnostic problem $\wp(\mathcal{O}, \mathfrak{M})$ is transformed by the diagnostic engine into a surrogate $\wp(\mathcal{I}(\mathcal{O}), \mathcal{M}(\mathfrak{M}))$. As for the index space, the model space is made up incrementally, following a *lazy evaluation* approach: the model space is extended only when necessary for the diagnostic engine. Essentially, a model space is a graph where nodes correspond to possible system states, while edges are marked by visible labels. Intuitively, a transition between nodes of the model space, $N \xrightarrow{\ell} N'$, occurs when the new observation fragment involves label ℓ. Both nodes and edges of the model space carry compiled diagnostic information: the dynamic solution of the diagnostic problem can be generated based on such information provided the index space is somehow linked to the model space. Specifically, each node $\Im$ of $\mathcal{I}(\mathcal{O})$ must be *decorated* with the set of model-space states which comply with all the paths up to $\Im$ in $\mathcal{I}(\mathcal{O})$. Such a decoration is grounded on the common alphabet of the regular language of $\mathcal{I}(\mathcal{O})$ and of $\mathcal{M}(\mathfrak{M})$, namely the domain of visible labels. For example, if $\langle \ell_1, \ldots, \ell_k \rangle$ is a string of the language of $\mathcal{I}(\mathcal{O})$ ending at node $\Im$, and the same sequence of labels is also a string of the language of $\mathcal{M}(\mathfrak{M})$ ending at node N, then the decoration of $\Im$ will include N. Since several different strings may end at $\Im$, the decoration of $\Im$ will include several nodes of $\mathcal{M}(\mathfrak{M})$. Accordingly, the *Increment* algorithm has been extended to cope with decorated index spaces too.

7 Conclusion

Both the observation graph and the index space are modeling primitives for representing temporal observations. Whereas the observation graph is the front-end representation, suitable for modeling an observation while it is being received over a time interval, the index space is a back-end representation, suitable for model-based problem-solving

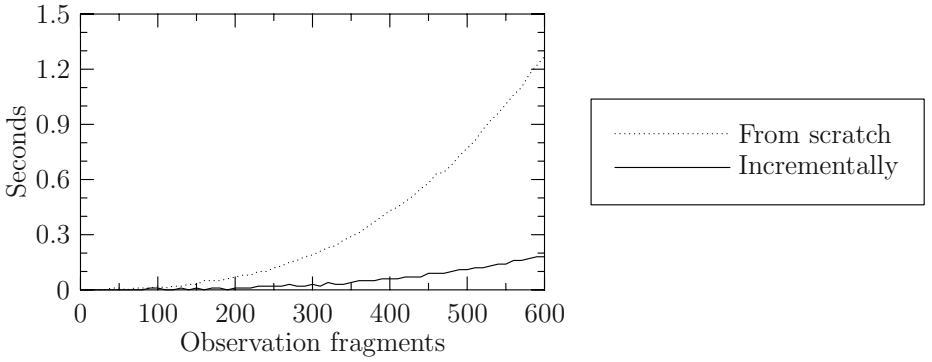


Fig. 4. Experimental results: index-space computation-time (y-axis) vs. number of observation fragments (x-axis)

and as a standard interchange format of uncertain observations among distinct application contexts. This paper has presented a technique for constructing the index space incrementally, while receiving observation fragments one at a time. This is significant whenever a nonmonotonic processing step has to be performed after each observation fragment is received, as is when the tasks of supervision and dynamic diagnosis (and state estimation, in general), are considered. The *Increment* algorithm is an attempt to achieve the stated goals. Experimental results, shown in Fig. 4, highlight that the algorithm (implemented in C language) is efficient. The diagram shows the time (in seconds) to compute the index space of an observation composed of (up to) 600 fragments. The curve on the top is relevant to the computation of each index space from scratch. The curve on the bottom corresponds to the incremental computation. The research still needs to perform computational analysis, and to gather further experimental results based on observations with different sizes and uncertainty degrees.

References

1. Rozé, L.: Supervision of telecommunication network: a diagnoser approach. In: DX 1997. Eighth International Workshop on Principles of Diagnosis, Mont St. Michel, F, pp. 103–111 (1997)
2. Brusoni, V., Console, L., Terenziani, P., Dupré, D.T.: A spectrum of definitions for temporal model-based diagnosis. Artificial Intelligence 102(1), 39–80 (1998)
3. Baroni, P., Canzi, U., Guida, G.: Fault diagnosis through history reconstruction: an application to power transmission networks. Expert Systems with Applications 12(1), 37–52 (1997)
4. Baroni, P., Lamperti, G., Pogliano, P., Zanella, M.: Diagnosis of large active systems. Artificial Intelligence 110(1), 135–183 (1999)
5. Wotawa, F.: On the relationship between model-based debugging and program slicing. Artificial Intelligence 135(1-2), 125–143 (2002)
6. Köb, D., Wotawa, F.: Introducing alias information into model-based debugging. In: Carcassonne, F. (ed.) DX 2004. Fifteenth International Workshop on Principles of Diagnosis, pp. 93–98 (2004)
7. Mozetič, I.: Hierarchical model-based diagnosis. International Journal of Man-Machine Studies 35(3), 329–362 (1991)

8. Lamperti, G., Zanella, M.: Diagnosis of discrete-event systems from uncertain temporal observations. Artificial Intelligence 137(1-2), 91–163 (2002)
9. Hopcroft, J., Motwani, R., Ullman, J.: Introduction to Automata Theory, Languages, and Computation, 3rd edn. Addison-Wesley, Reading, MA (2006)
10. Lamperti, G., Zanella, M.: A bridged diagnostic method for the monitoring of polymorphic discrete-event systems. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 34(5), 2222–2244 (2004)
11. Lamperti, G., Zanella, M.: Dynamic diagnosis of active systems with fragmented observations. In: Porto, P. (ed.) ICEIS 2004. Sixth International Conference on Enterprise Information Systems, pp. 249–261 (2004)
12. Lamperti, G., Zanella, M.: Diagnosis of Active Systems – Principles and Techniques. The Kluwer International Series in Engineering and Computer Science, vol. 741. Kluwer Academic Publisher, Dordrecht, NL (2003)
13. Lamperti, G., Zanella, M.: Uncertain temporal observations in diagnosis. In: Berlin, D. (ed.) ECAI 2000. Fourteenth European Conference on Artificial Intelligence, pp. 151–155. IOS Press, Amsterdam, NL (2000)

Towards a Fuzzy Ontology Definition and a Fuzzy Extension of an Ontology Editor

Silvia Calegari and Davide Ciucci

Dipartimento di Informatica Sistemistica e Comunicazione
Università di Milano – Bicocca
Viale Sarca, 336/14, I–20126 Milano, Italia
{calegari,ciucci}@disco.unimib.it

Abstract. This paper presents an enrichment of classical computational ontologies with fuzzy logic to create fuzzy ontologies. It is a step towards managing vagueness and facing the nuances of natural languages in ontology-based applications. Our proposal is implemented in the KAON ontology editor, that allows to handle ontology concepts in a high-level environment.

Keywords: Ontology, fuzzy logic, KAON, concept modifiers.

1 Introduction

An ontology is a formal conceptualization of a particular domain of interest shared among heterogeneous applications. It consists of *entities, attributes, relationships* and *axioms* to provide a common understanding of the real world [1,2,3]. With the support of ontologies, users and systems can communicate with each other through an easier information exchange and integration [4]. Ontologies help people and machines to communicate concisely by supporting information exchange based on semantics rather than just syntax.

There are ontological applications where information is often vague and imprecise. For instance, the semantic-based applications of the Semantic Web [5], such as e-commerce, knowledge management, web portals, etc. However, the conceptual formalism supported by a typical ontology may not be sufficient to represent vague information that is commonly found in many application domains.

A possible solution to tackle these problems, is to incorporate fuzzy logic into ontologies. Indeed, the aim of fuzzy set theory [6] introduced by L. A. Zadeh [7] is to describe vague concepts through a generalized notion of set, according to which an object may belong to a certain degree (typically a real number from the interval [0,1]) to a set. For instance, the semantic content of a statement like "Cabernet is a deep red acidic wine" might have degree, or truth-value, of 0.6. Up to now, fuzzy sets and ontologies are jointly used to resolve vague information problems in various areas, for example, in text retrieval [8,9,10] or to generate a scholarly ontology from a database in ESKIMO [11] and FOGA [12] frameworks. However, in none of these examples there is a fusion of fuzzy set theory with ontologies.

The aim of this paper is to present a proposal to directly integrate fuzzy logic in ontology in order to obtain an extension of the ontology that is more suitable for solving

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 147–158, 2008.

reasoning problems with vague knowledge. It is a first step towards the realization of a theoretical model and of a complete framework based on ontologies that are able to consider the nuances of natural languages. In literature, an extension of the domain ontology with fuzzy concept has been introduced in [13], however only for Chinese news summarization. Recently in [14], a first non-complete real formalization of a fuzzy ontology is given.

Moreover, in this paper we address the problem on how to insert fuzzy logic in KAON [15]. This software consists in a number of different modules providing a broad range of functionalities centered around creation, storage, retrieval, maintenance and application of ontologies. KAON allows the use of an ontology at high-level, and the relative conceptual models are defined in a natural and easily understandable way.

The rest of the paper is organized as follows: Section 2 defines a fuzzy ontology and explains how to define and use fuzzy values in it. Section 3 presents the ontology editor used and it is shown how to integrate it with our framework.

2 Definition and Use of a Fuzzy Ontology

In this section, we present a framework to support and reason with uncertainty. This is a focus aspect for all ontology-based applications where the user is interested in information that often contains imprecise and vague description of concepts. For example, one may be interested in finding "a *very strong* flavored red wine" or in reasoning with concepts such as "a *cold* place", "an *expensive* item", "a *fast* motorcycle", etc.

In order to face these problems the proposed approach is based on fuzzy sets theory. Our aim is to extend an ontology editor to directly handle uncertainty during the ontology definition, so that to enrich the knowledge domain.

At first, let us remind the definition of a fuzzy set. Let us consider a nonempty set of objects U, called the *universe*. A *fuzzy set* or *generalized characteristic functional* is defined as a $[0,1]$-valued function on U, $f : U \mapsto [0,1]$. Given an object $x \in U$, $f(x)$ represents the *membership value* of x to the set f. In the following of this section we explain how to introduce fuzzy values on different objects of an ontology and how to automatically correct them. Finally, we give some hint on the possible applications of a fuzzy ontology.

2.1 Defining a Fuzzy Value

The first problem to tackle is how to assign a fuzzy value to an entity of the ontology. The trade off is between understandability and precision, since [16]

> to obtain high degree of interpretability and accuracy is a contradictory purpose and, in practice, one of the two properties prevails over the other one. Depending on what requirement is mainly pursued, the Fuzzy Modelling field may be divided into two different areas:
>
> 1. Linguistic fuzzy modelling – The main objective is to obtain fuzzy models with a good interpretability
> 2. Precise fuzzy modelling – The main objective is to obtain fuzzy models with a good accuracy.

Since our goal is to be as general as possible, both the possibilities are given to the expert: define a precise value or a linguistic one. In the former case the expert, while creating the ontology, defines each relation as among concepts as a fuzzy relation. So, for example, in a cats ontology, the fact that "Garfield has sense of humour with value 0.9" can be represented as the fuzzy taxonomic relation $\mathcal{T}$ between the concepts Garfield and Sense-of-Humour: $\mathcal{T}$(Sense-of-humour,Garfield) = 0.9. Or, in a wine ontology, the relation Taste(wine, full-bodied) = 0.4 means that the concept *wine* has a *full-bodied* (the value) *taste* (the property) with degree *0.4*.

On the other hand, the second possibility is to choose as membership value, a *label* in a given set. We have chosen the set L ={little, enough, moderately, quite, very, totally} which is clearly not exhaustive of all the possible labels, but it can intuitively be modified as desidered.

In this case, the numeric value corresponding to each label is automatically assigned according to Table 1.

Table 1. Assignement of fuzzy value to labels

Label	Value
little	0.2
enough	0.4
moderately	0.6
quite	0.7
very	0.8
totally	1

Summing up, we give the chance to add a membership value to a pair (concept / instance, property) in two different ways: through a precise value $v \in [0, 1]$ or choosing a label in the predefined set L. Thus, through the function g we define a new relation in the ontology domain.

Let us remark that the fuzzy value assigned using one of the two methods is a number in the unit interval $[0, 1]$, that is, the usual support of a many valued logic. Hence, applications based on fuzzy ontologies can use this value taking advantage of standard and well-studied tools. For instance, in order to put together two (or more) different fuzzy values, an *aggregation operator* [17] can be used. Typical examples are *t-norm* and *t-conorms* [18], that is, binary mappings which give a semantic to the "OR", "AND" operators. The most known are Gödel norm and conorm, i.e., the min–max operators. Considering the above example, it may be necessary to compute the truth value of the statement "Garfield is orange AND has sense of humour". If it is known that hasColor(Garfield,orange)= 0.8 and $\mathcal{T}$(Sense-of-Humour,Garfield) = 0.9 then [HasColor(Garfield,orange) AND $\mathcal{T}$(Sense-of-humour,Garfield)] = min$\{0.8, 0.9\} = 0.8$.

Finally, we can give the definition of fuzzy ontology.

Definition 1. *A* Fuzzy Ontology *is defined as the tuple* $\mathbf{O_F} = \{\mathbf{C}, \mathbf{R}, \mathbf{F}, \mathbf{A}\}$ *where:*

- $\mathbf{C}$ *is a set of concepts.*
- $\mathbf{R}$ *is a set of relations. Each* $R \in \mathbf{R}$ *is a n-ary fuzzy relation on the domain of concepts,* $R : \mathbf{C}^n \mapsto [0, 1]$. *A special role is held by the taxonomic relation* $\mathcal{T} : \mathbf{C}^2 \mapsto [0, 1]$ *which identifies the fuzzy subsumption relation among concepts.*

- $\mathbf{F}$ *is a set of the fuzzy relations on the set of concepts and a specific domain contained in* $\mathcal{D} = \{integer, string, ...\}$. *In detail, they are n-ary functions such that each element* $F \in \mathbf{F}$ *is a relation* $F : \mathbf{C}^{(n-1)} \times P \mapsto [0, 1]$ *where* $P \in \mathcal{D}$.
- $\mathbf{A}$ *is a set of axioms expressed in a proper logical language, i.e., predicates that constrain the meaning of concepts, relationships and functions.*

A Fuzzy Knowledge Base *is a tuple* $(\mathbf{O}_F, \mathcal{I})$ *where* $\mathbf{O}_F = (\mathbf{C}, \mathbf{R}, \mathbf{F}, \mathbf{A})$ *is a Fuzzy Ontology and* $\mathcal{I}$ *is the set of instances. Every concept* $C \in \mathbf{C}$ *is a fuzzy set on the domain of instances* $\mathcal{I}$, $C : \mathcal{I} \mapsto [0, 1]$. *Further, all the relations in* $\mathbf{R}$ *and* $\mathbf{F}$ *are extended to include instances. That is, we define the set of entities as* $\mathbf{E} = \mathbf{C} \cup \mathcal{I}$. *Then, all relations are now defined on* $\mathbf{E}$ *instead of simply on* $\mathbf{C}$.

2.2 Updating a Fuzzy Value

Once an expert has created a fuzzy ontology, it is not realistic to assume that it is perfect and that any fuzzy value is well-defined and suited to any context. Thus, a mechanism to change fuzzy values in order to fit them in the best way to a specific environment or, in general, to make them more correct is needed. Here, we propose a method to update fuzzy values according to results of some queries on some documents. We do not enter into details about how syntactically specify queries, but we assume that we are able to perform them and that their results are available to us.

Let us suppose that the current fuzzy value is f and as a result of a query it must be updated to f_{new}. The simplest possibility is to set $f := f_{new}$. However, it is reasonable to suppose that after some queries the fuzzy property has reached a stable value, hence it is not useful to change it with f_{new}, losing all the history of the acquired knowledge. A solution can be to diminish the importance of f_{new} at any change:

$$f := f + \frac{f_{new} - f}{Q + 1} \tag{1}$$

where Q is the number of updates perfomed for that value. Clearly, the value Q must be stored in the ontology for any defined fuzzy value.

Now, the issue is how to compute a new fuzzy value f_{new}. It is unlikely to find in a document a precise definition of a fuzzy value, but usually a linguistic qualifier can be found. For example, we do not find "Cabernet has a dry taste with value 0.6", but it make sense a document such "Cabernet has a very dry taste". So a method to make use of this kind of information is needed. Here we propose an approach based on concept modifiers [19].

A concept modifier has the effect to alter the fuzzy value of a property. Given a set of linguistic hedges such as "very", "more or less", "slighty", a concept modifier is a chain of one or more hedges, such as "very slightly" or "very very slightly". To any (linguistic) concept modifier it is necessary to associate a (numerical) membership modifier.

Definition 2. *A* membership modifier *is a value* $\beta > 0$ *which is used as an exponent to modify the value of a membership function* f *as* f^{β}.

According to their effect on a fuzzy value, a hedge can be classified in two groups: *concentration* type and *dilation* type. The effect of a concentration modifier is to reduce the grade of a membership value. Thus, in this case, it must be $\beta > 1$. For instance, to the hedge "very", it is usually assigned $\beta = 2$. So, if we know that $Taste(Cabernet, dry) = 0.8$, i.e., "Cabernet has a dry taste with value 0.8", then "Cabernet has a very dry taste" with value $0.8^2 = 0.64$. On the contrary a dilation hedge has the effect to raise a membership value, that is $\beta \in (0, 1)$. For instance, if to slightly it is assigned $\beta = 0.25$ and $Taste(Cabernet, dry) = 0.8$, then "Cabernet has a slightly dry taste" with value $0.8^{0.25} = 0.95$ according to the intuition that if something is "dry" then it is even more "slightly dry". Let us remark that this approach is different from the original Zadeh's one [19], where "slightly", as well as other modifiers, is handled in a more complicated manner. This method has the advantage to give a uniform and simply way to manage concept modifiers, even if, a deeper study about the semantic of this way of handling chain of modifiers is needed.

As can be seen, a concept modifier is used in literature to define a new fuzzy membership given an already existing one. For example, if we know the fuzzy value of property *red* we can infer the fuzzy membership of property *very red* simply by raising to the power 2 the value of red (see the above examples). However, here we are in the opposite situation. As an example, let us suppose to know, from an ontology, the red property and also, as a result of a query, that a certain object (here o) is "very red". Hence, from very red we need to infer a new red property (before "red" was fixed, here it changes) for that object and clearly if an object is very red it is even more red. So, if in the ontology $HasColor(o, red) = 0.7$, we must increase this value, for example $HasColor(o, red) = 0.7^{0.5}$. In conclusion, the effect of very is to raise the value of the property it is referred to and not to reduce its magnitude. In a schematic way, it is possible to say that in the usual case it is performed the deduction

$$\text{red} \rightarrow \text{very red}$$

whereas in this situation:

$$\text{red and very red} \rightarrow \text{red}$$

This argument also applies to all the other concept modifiers. Thus, in our case what is usually considered as a concentration modifier becomes a dilation one and vice versa.

Two issues need now to be faced: decide which modifiers we consider (and which are their values) and define a method to compute β values for chains of concept modifiers.

About the former problem, the chosen set of hedges is H={very, much more, more, more or less, moderate, slightly}. We remark that this is only one of the possible choices. The set H can be changed according to one's needs, on condition that it satisfies the following two properties.

- The set H is totally ordered, i.e., very $< \ldots <$ slightly and only the β value for the smaller and greater elements are fixed.
- The two subsets of contraction and dilation hedges, in our case {more or less, moderate, slightly} and {very, much more, more} have the same cardinality.

These two conditions are due to the fact that we propose to adopt the algorithm presented in [20] in order to compute the membership modifier of a sequence of hedges.

The only difference is the order inversion of the hedges, this is due to the use we are doing of concept modifiers which is, as explained above, opposite to the original approach.

Moreover, as a further modifier, we consider also the negation connective *not*. Special attention must be kept in managing this modifier since it is not possible to find a general way to give a semantic to negations, specially when used in combination with other modifiers. As an example, let us think to the sentence "Cabernet has not a dry taste". If it is known that "Cabernet has a dry taste" with value 0.8, then we can apply the standard semantic of fuzzy negation and derive that "Cabernet has not a dry taste" with value $1-0.8=0.2$. On the other hand, if we find that "Cabernet has not a very dry taste", and follow the same semantic, we have $1-\sqrt{0.8}=0.11$. However, "not very dry " induces to think to something which is dry but not at an high degree (say something around 0.6 or 0.7) and this is not correctly mirrored by the value 0.11. In [9] a solution to integrate *not* in the set H of all concept modifiers is presented. This solution does not seem to us a good one, since it cannot be applied directly to a property, but only to another concept modifier and also in this case it can generate a negative β if the original algorithm of [20] is not modified. Thus, in [21] we tackle this problem and propose two different solutions. The first one is based on the model developed by Pacholczyk in [22,23,24,25], which allows to establish the meaning of a negation according to the context, but only on a fixed set of modifiers. We modify it in order to to be able to correctly interpreting a non-predetermined combination of modifiers. For example, in our proposal, we can handle a chain of modifiers, such as "little, very very, very very little" and so on. The main idea behind the second model is to distinguish the way a negation is used inside a sentence: to alter the meaning of a property or the meaning of a linguistic modifier, and to handle the two uses differently.

Another problem, which remains open, is that the set H of concept modifiers is certainly not exhaustive of all the nuances of natural language. Lots of elements could be added to H and this will require new algorithms to handle it, since not all existing concept modifiers can be totally ordered or exactly split into two subgroups of same cardinality.

2.3 An Example of Application: Extending Queries

In this section we give an example of a possible use of fuzzy ontologies.

When performing a query on a document, it is a usual practice to extend the set of concepts already present in the query with other ones which can be derived from an ontology. Tipically, given a concept, also its parents and children can be added to the query and then searched in the document.

A possible use of fuzzy ontology is to extend queries with, besides children and parents, instances of concepts which satisfies to a certain degree the query. Let us explain it with an example. We are given a clothes ontology and a query looking for "a very long and black coat". In the ontology there are two instances of coat: X which has property "long" with value 0.7 and Y which has property "long" with value 0.3. Thus, it is natural to extend the original query adding, not only parents and children of the concept "coat", but also the instance X, because "$long=0.7$" can be interpreted as "very long". On the other hand, the instance Y is not added to the extended query since "$long=0.3$" does not mean "very long".

To make a choice on which instances have to be added to the extended query, we have to decide how linguistic labels are mapped to numerical values. The solution is again as in Section 2.1, that is only label belonging to set L are admitted in queries and they are mapped to numerical values according to Table 1. If c is a concept, p is a property and l a label then $\mu(c, p, l)$ is the value given to the label l for property p and concept c. For instance in the above example, the property "a very long coat" is translated to μ(coat, long, very) $= 0.8$. Now, we consider all the instances i of the concept c and they are included in the extended query if and only if :

$$|\mu(c, p, l) - p(i)| \leq \epsilon \tag{2}$$

where $\epsilon \in [0, 1]$ is a level of tolerance. Obviously, the number of instances to be added to the extended query depends on the value of ϵ, the greater is ϵ the most are the instances. The boundary cases are $\epsilon = 0$, only the instances that exactly match the query are included, and $\epsilon = 1$, all the instances are included. Coming back to the example, if we fix $\epsilon = 0.2$, then, $|\mu(coat, long, very) - long(X)| = 0.8 - 0.7 = 0.1 \leq 0.2 = \epsilon$, whereas $|\mu(coat, long, very) - long(Y)| = 0.8 - 0.3 = 0.5 \geq 0.2 = \epsilon$. Hence, X is included in the extended query and Y is not.

Clearly, this is the simplest case where only one property is present in the query. If two or more request must be satisfied, a generalization of equation (2) is needed. Let us suppose that in the query there are n properties referred to the same concept c, then instance i is considered iff

$$\frac{\sum_{j=1}^{n} |\mu(c, p_j, l_j) - p_j(i)|}{n} \leq \epsilon \tag{3}$$

That is we require that the mean value of the distances between the values of the properties in the query and the values of the properties in the instance is less (or equal) than the tolerance ϵ.

3 Adding Fuzziness in KAON

The KAON project is a meta-project carried out at the Institute AIFB, University of Karlsruhe and at the Research Center for Information Technologies (FZI).

KAON includes a comprehensive tool suite allowing easy creation, maintenance and management of ontologies. Furthermore, it provides a framework for building ontology-based applications. An important user-level application supplied by KAON is an ontology editor called OI-modeler. It allows to handle entities of an ontology in a natural way. The most important features of the OI-modeler are its support for manipulation of large ontologies and the support for user-directed evolution of ontologies. Ontologies can be simultaneously edited by multiple users [26]. The ontology navigation is more easy through graph-based and tree-based metaphors.

In the last years, KAON has been used in many areas like e-commerce and b2b applications [27], autonomic and self-healing, self-configuring computational system [28] and more recently, it has been applied to the Semantic Web [29,30].

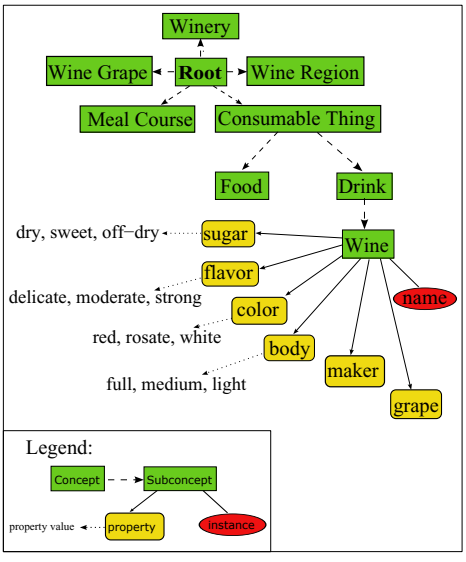


Fig. 1. Wine ontology

3.1 Ontologies in KAON

An ontology in KAON consists of concepts (sets of elements), properties (specifications of how objects may be connected) and instances grouped in reusable units called OI-models (ontology-instance models) [26]. The conceptual model proposed allows to define an entity in different ways, depending on the point of view of the observer. That is, an entity can be interpreted as a concept, as well as an instance. Moreover, property instantiation must be in accordance with the domain and range constraints (i.e. axioms, general rules, value-allowed) and must obey the cardinality constraints, as specified by the property specifications. An OI-model may include other OI-models, and have immediate access to all definitions from the included model. Figure 1 is an example of an ontology in KAON: it represents only a partial ontology definition about wine.

In KAON language, it is possible to define well-known symmetric, transitive and inverse properties, with the addition of modularization and meta-modeling [26]. Obviously, each of these features allow to manage two types of implicit knowledge: axioms and general rules. The formers are a standard set of rules, such as the transitive property, the latters furnish a way to define rules to constrain information defined in a specific ontology domain.

Moreover, KAON language allows to specify so-called lexical entries (i.e. labels, synonyms, stems, or textual documentation) which reflect various nuances of natural languages. For example, the same lexical entry may be associated with several elements: the label BEAR may be associated with an instance representing a bear as an animal or as a puppet. Furthermore, the instances can be defined in different languages, namely English, German, French, Spanish, Portuguese, Arabic and Chinese.

3.2 Fuzzy Ontologies in KAON

Our aim is to enrich KAON language adding the proposed fuzzy-sets approach. In the following, it is showed how we have integrated our framework in the KAON project.

Figure 2 represents a "Fuzzy Inspector" developed to create in KAON the fuzzy ontology. The new panel has been called "Fuzzy Logic" (see right lower corner in Figure 2) and it allows the expert an easy fuzzy logic integration. The Fuzzy Inspector is composed by a table representing fuzzy entity, membership degree and number of updates Q.

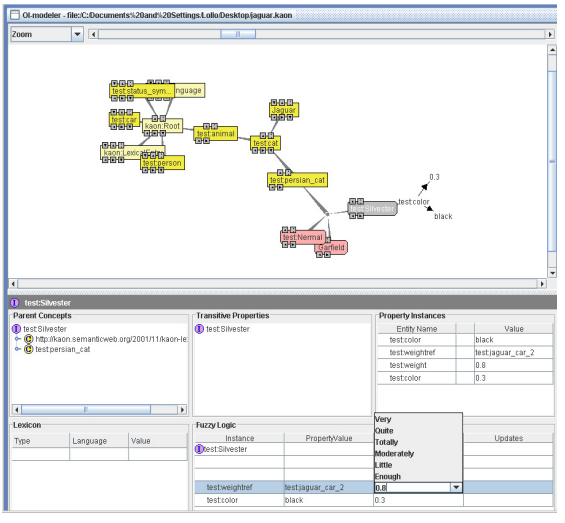


Fig. 2. New KAON ontology overview

The domain expert can choose his fuzzy entity (concept or instance) simply clicking up through two types of interface proposed by KAON, namely graph-based and tree-based metaphors. Moreover, the expert can select the link between an entity and a property adding its fuzzy value.

Thus, the expert can create a fuzzy ontology directly inserting fuzzy values through the Fuzzy Inspector Panel purposely developed. In Figure 2, we propose two ways to use the Fuzzy Logic Panel. In the first row, the expert types the membership degree according to his point of view. In the second row, he can choose the apposite value by a list. The selected element in the list will be referred to an a-priori defined numerical value as explained in Section 2.1. In the definition phase of the ontology the number of the updates is zero. This value will be changed during the queries in accordance with the functions defined in previous sections.

3.3 Fuzzy OWL

KAON's ontology language is based on RDFS [31] with proprietary extensions for property characteristics (symmetric, transitive and inverse), cardinality, modularization, meta-modelling and explicit representation of lexical information. In literature, all the

limits about the RDFS are well-known. Thus, the inference engine KAON2 [32] has been developed. The main difference with respect to the previous KAON version is the supported ontology language: KAON used a proprietary extension of RDFS, whereas KAON2 is based on OWL DL [33], whose theoretical counterpart is the description logic $\mathcal{SHOIN}(\mathcal{D})$.

Recently, some proposals to integrate fuzzy logic in Description Logic and OWL have been presented [34,35,36]. In [37,38], we defined a fuzzy extension of the OWL language considering fuzzy $\mathcal{SHOIN}(\mathcal{D})$. We have extended the syntax and semantic of fuzzy $\mathcal{SHOIN}(\mathcal{D})$ giving a complete formalization of fuzzy axioms and introducing some new features. In particular, we added the possibility to have fuzzy relations with modifiers, and not only modified fuzzy concepts. This can be helpful to define a new type of non-taxonomic fuzzy relationship, named *correlation*, in order to establish a semantic link between entities of the fuzzy ontology [39]. Further, we gave a new possibility to define a concept: $\leq_\alpha C$ (and similarly $\geq_\alpha$, $<_\alpha$, $>_\alpha$) which enable us to define, for instance, the fuzzy set of "people which are tall with value lower than 0.3" or the "wines which have a dry taste with a value at least of 0.6". Finally, we gave a new semantic of cardinality restriction.

Our idea is to map the fuzzy ontology definition into the corresponding Fuzzy-OWL language through the syntax and semantic of fuzzy $\mathcal{SHOIN}(\mathcal{D})$. The last step to complete this work is a theoretical analysis of decidability and computability issues of fuzzy $\mathcal{SHOIN}(\mathcal{D})$ and the integration of fuzzy OWL in an extension of KAON2.

4 Conclusions

In this paper, we introduced fuzzy logic directly in the ontology, enriching the actual features proposed by classical ontologies. The proposed solution allows to represent and to reason with vague information, a delicate problem for all those areas where the applications are based on ontology. The domain expert has two possibilities to add a membership value in an ontology domain: through a pair *(*{*concept,instance*}*,property)* or through an entity {*concept, instance*}. In both solutions, he/she can assign this degree through a precise value $v \in [0, 1]$ or choosing a label in the predefined set L.

We have also proposed a method, based on concept modifiers, to automatically update the membership degree during queries, useful, as example, for the extraction of more relevant documents.

Further, we have integrated fuzzy logic in the ontology editor KAON, developing a suitable Fuzzy Inspector Panel.

Due to all the limits of KAON and RDFS, in the future, it will be necessary to implement a fuzzy extension of OWL in KAON2 and, finally, to test all the proposed framework.

References

1. Lammari, N., Métais, E.: Building and maintaining ontologies: a set of algorithms. Data and Knowledge Engineering 48, 155–176 (2004)
2. Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5, 199–220 (1993)

3. Guarino, N., Giaretta, P.: Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: Mars, N. (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press, Amsterdam (1995)
4. Soo, V.W., Lin, C.Y.: Ontology-based information retrieval in a multi-agent system for digital library. In: 6th Conference on Artificial Intelligence and Applications, pp. 241–246 (2001)
5. Berners-Lee, T., Hendler, T., Lassila, J.: The semantic web. Scientific American 284, 34–43 (2001)
6. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, Englewood Cliffs (1995)
7. Zadeh, L.A.: Fuzzy sets. Inform and Control 8, 338–353 (1965)
8. Bouquet, P., Euzenat, J., Franconi, E., Serafini, L., Stamou, G., Tessaris, S.: Specification of a common framework for characterizing alignment. IST Knowledge web NoE 2.2.1 (2004)
9. Singh, S., Dey, L., Abulaish, M.: A Framework for Extending Fuzzy Description Logic to Ontology based Document Processing. In: Favela, J., Menasalvas, E., Chávez, E. (eds.) AWIC 2004. LNCS (LNAI), vol. 3034, pp. 95–104. Springer, Heidelberg (2004)
10. Abulaish, M., Dey, L.: Ontology Based Fuzzy Deductive System to Handle Imprecise Knowledge. In: InTech 2003. Proceedings of the 4th International Conference on Intelligent Technologies, pp. 271–278 (2003)
11. Matheus, C.: Using Ontology-based Rules for Situation Awareness and Information Fusion. In: Position Paper presented at the W3C Workshop on Rule Languages for Interoperability (2005)
12. Quan, T., Hui, S., Cao, T.: FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web. In: Knowledge Discovery and Ontologies (KDO-2004), Workshop at ECML/PKDD (2004)
13. Chang-Shing, L., Zhi-Wei, J., Lin-Kai, H.: A fuzzy ontology and its application to news summarization. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 35, 859–880 (2005)
14. Sanchez, E., Yamanoi, T.: Fuzzy ontologies for the semantic web. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 691–699. Springer, Heidelberg (2006)
15. KAON: Karlsruhe Ontology and Semantic Web Tool Suite (2005), http://kaon.semanticweb.org
16. Casillas, J., Cordon, O., Herrera, F., Magdalena, L.: Accuracy improvements to find the balance interpretability-accuracy in linguistic fuzzy modeling:an overview. In: Accuracy Improvements in Linguistic Fuzzy Modeling, pp. 3–24. Physica-Verlag, Heidelberg (2003)
17. Calvo, T., Mayor, G., Mesiar, R. (eds.): Aggregation Operators. Physica–Verlag, Heidelberg (2002)
18. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer Academic, Dordrecht (2000)
19. Zadeh, L.A.: A fuzzy-set-theoretic interpretation of linguistic hedges. Journal of Cybernetics 2, 4–34 (1972)
20. Khang, T.D., Störr, H., Hölldobler, S.: A fuzzy description logic with hedges as concept modifiers. In: Third International Conference on Intelligent Technologies and Third Vietnam-Japan Symposium on Fuzzy Systems and Applications, pp. 25–34 (2002)
21. Calegari, S., Radaelli, P., Ciucci, D.: Two fuzzy-set models for the semantic of linguistic negations. In: Studies in fuzziness and Soft Computing, Springer, Heidelberg (to appear)
22. Pacholczyk, D.: A new approach to linguistic negation of nuanced information in knowledge-based systems. In: Giunchiglia, F. (ed.) AIMSA 1998. LNCS (LNAI), vol. 1480, pp. 363–376. Springer, Heidelberg (1998)
23. Pacholczyk, D.: A new approach to linguistic negation based upon compatibility level and tolerance threshold. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, Springer, Heidelberg (1998)

24. Pacholczyk, D., Hunter, A.: An extension of a linguistic negation model allowing us to deny nuanced property combinations. In: Hunter, A., Parsons, S. (eds.) ECSQARU 1999. LNCS (LNAI), vol. 1638, pp. 316–327. Springer, Heidelberg (1999)
25. Pacholczyk, D., Levrat, B.: Coping with linguistically denied nuanced properties: a matter of fuzziness and scope. In: Proceeding of ISIC, pp. 753–758. IEEE, Los Alamitos (1998)
26. AA.VV.: Developer's Guide for KAON 1.2.7. Technical report, FZI Research Center for Information and WBS Knowledge Management Group (2004)
27. Motik, B., Maedche, A., Volz, R.: A Conceptual Modeling Approach for Semantics-Driven Enterprise Applications. In: Meersman, R., Tari, Z., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 1082–1099. Springer, Heidelberg (2002)
28. Stojanovic, L., Schneider, J., Maedche, A., Libischer, S., Studer, R., Lumpp, T., Abecker, A., Breiter, G., Dinger, J.: The role of ontologies in autonomic computing systems. IBM Systems Journal 43, 598–616 (2004)
29. Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Sure, Y., Tane, J., Volz, R., Zacharias, V.: KAON - Towards a large scale Semantic Web. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) EC-Web 2002. LNCS, vol. 2455, pp. 304–313. Springer, Heidelberg (2002)
30. Oberle, D., Staab, S., Studer, R., Volz, R.: Supporting application development in the semantic web. ACM Trans. Inter. Tech. 5, 328–358 (2005)
31. RDFS: Resource Description Framework Schema (2004), http://www.w3.org/TR/PR-rdf-schema
32. KAON2: Karlsruhe Ontology and Semantic Web Tool Suite 2 (2005), http://kaon2.semanticweb.org
33. OWL: Ontology Web Language (2004), http://www.w3.org/2004/OWL/
34. Hajek, P.: Making fuzzy description logic more general. Fuzzy Sets and Systems 154, 1–15 (2005)
35. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrocks, I.: Fuzzy OWL: Uncertainty and the Semantic Web. In: OWL-ED 2005. International Workshop of OWL: Experiences and Directions, Galway, Ireland (2005)
36. Straccia, U.: A fuzzy description logic for the semantic web. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web. Capturing Intelligence, pp. 73–90. Elsevier, Amsterdam (2006)
37. Calegari, S., Ciucci, D.: Fuzzy Ontology, Fuzzy Description Logics and Fuzzy-OWL. In: Masulli, F., Mitra, S., Pasi, G. (eds.) WILF 2007. LNCS (LNAI), vol. 4578, pp. 118–126. Springer, Heidelberg (2007)
38. Calegari, S., Ciucci, D.: Fuzzy Ontology and Fuzzy-OWL in the KAON Project. In: Fuzz-IEEE Proceedings 2007, pp. 1415–1420 (2007)
39. Calegari, S., Farina, F.: Fuzzy Ontologies and Scale-free Networks Analysis. International Journal of Computer Science and Applications IV, 125–144 (2007)

Retrieval of Collaborative Filtering Nearest Neighbors in a Content-Addressable Space

Shlomo Berkovsky, Yaniv Eytani, and Larry Manevitz

Computer Science Department, University of Haifa, 31905, Haifa, Israel
{slavax,ieytani,manevitz}@cs.haifa.ac.il

Abstract. Collaborative Filtering (*CF*) is considered one of the popular and most widely used recommendation techniques. It is aimed at generating personalized item recommendations for the users based on the assumption that similar users have similar preferences and like similar items. One of the major drawbacks of the CF is its limited scalability, as the CF computational effort increases linearly with the number of users and items. This work presents a novel variant of the CF, employed over a content-addressable space. This heuristically decreases the computational effort required by the CF by restricting the nearest neighbors search applied by the CF to a set potentially highly similar users. Experimental evaluation demonstrates that the proposed approach is capable of generating accurate recommendations, while significantly improving the performance in comparison with the traditional implementation of the CF.

1 Introduction

Recommender Systems [16] are one of the commonly used approaches to address the Information Overloading problem. They systems assist a user in selecting a suitable item among a set of potentially selectable items by predicting the user's opinion on the items [19]. Currently, Recommender Systems are used in a variety of application domains, such as movies [7], jokes [6], and music [1], and they exploit several recommendation techniques, such as Collaborative [9] and Content-Based Filtering [12], Case-Based Reasoning [17] and many hybrid techniques [4].

Collaborative Filtering (CF) is probably the most familiar and one of the most widely-used techniques to generate predictions in Recommender Systems. It relies on the assumption that people who agreed in the past will also agree in the future [21]. The input for the CF algorithm is a matrix of users' ratings on a set of items, where each row represents the ratings provided by a single user and each column represents the ratings provided by different users on a single item. CF aggregates the ratings to recognize similarities between users and generates the prediction for an item by weighting the ratings of similar users on this item.

The CF algorithm is typically partitioned to three generic stages: (1) Similarity Computation: weighting all the users with respect to their similarity with the *active user* (i.e., the user, whose ratings are being predicted), (2) Neighborhood Formation: selecting the most similar users for the prediction generation, and (3) Prediction Generation: computing the prediction by weighting the ratings of the selected users.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 159–178, 2008.

One of the major drawbacks of the CF is its limited scalability. The stages of Similarity Computation and Neighborhood Formation require comparing the active users with all the other users over all the available ratings. Hence, the computational effort required by the CF grows linearly with the number of users and the number of items in the ratings matrix. Thus, for a matrix containing ratings of *M* users on *N* items, the required computational effort is *O(MN)*. This poses a problem in systems, where the predictions are generated using millions of ratings on thousands of items, e.g., in Web-based Recommender Systems. Previous studies, (e.g., [3], [6], [5], and others) tackle the issue of reducing the computational effort required by the CF either by preprocessing of the ratings matrix or by distributing the computational stages. Nonetheless it remains one of the most important issues in the CF research community.

In this work we develop a fast heuristic variant of the CF algorithm that decreases the computational effort required by the Similarity Computation and the Neighborhood Formation stages. The basic assumption of the proposed heuristic algorithm is that losing general completeness of the exhaustive search (1) has a minor negative effect on the accuracy of the predictions, but (2) significantly decreases the required computational effort. Thus it provides a scalable approach, applicable to real-life scenarios with a high number of users and items, such as in Web-based systems.

The proposed heuristic approach is based on a notion of content-addressable data management [15] that provides an adaptive topology for mapping of users' profiles to a multi-dimensional space. This mapping implicitly clusters similar users and limits the Similarity Computation and the Neighborhood Formation stages to a heuristic search among the users that are potentially highly similar to the active user.

Experimental evaluation of the proposed approach demonstrates both high efficiency and good accuracy of the proposed algorithm in comparison with the traditional (exhaustive) *K-Nearest Neighbors* (KNN) search of the Neighborhood Formation stage. The evaluation also demonstrates that the algorithm is highly scalable with the number of nearest neighbors to be retrieved.

The rest of the paper is organized as follows. Section *2* describes the CF personalization technique and surveys the studies on the required computational effort reduction. Section *3* describes the CAN, a Peer-to-Peer content-addressable platform for decentralized data management. Section *4* describes the decentralized storage of users' profiles over the CAN platform and elaborates on the proposed heuristic variant of the CF over CAN. Section *5* presents and analyzes the experimental results. Finally, section *6* lists our conclusions and presents some open questions for future research.

2 Collaborative Filtering

Collaborative Filtering (*CF*) is probably one of the most familiar and widely-used recommendation techniques. An input for the CF is the so-called *ratings matrix*, where each user is represented by a set of explicit ratings given on various items, and each item is represented by a set of ratings given by the users. CF requires a similarity metric between users to be explicitly defined. The state-of-the-art CF systems exploit three similarity metrics: Cosine Similarity [7], Mean Squared Difference (MSD) [13], and Pearson correlation [19]. This work focuses on the MSD, computing the degree of similarity between users *x* and *y* by:

$$sim_{x,y} = \frac{\sum_{i=1}^{|x \cap y|} (R_{x,i} - R_{y,i})^2}{|x \cap y|}$$

where $|x \cap y|$ denotes the number of items rated by both users (typically, above some minimal threshold), and $R_{x,i}$ denotes the rating of user x on item i. In some sense, $sim_{x,y}$ can be considered as the *dissimilarity* of the users, as the lower the result of the MSD computation, the greater is the real similarity between the users. Prediction $P_{a,j}$ for the rating of the user a on item j is computed as a weighted average of the ratings of his/her K most similar users, i.e., *K nearest neighbors*, by:

$$P_{a,j} = R_a^{'} + \frac{\sum_{k=1}^{K} (R_{k,j} - R_k^{'}) \cdot sim_{a,k}}{\sum_{k=1}^{K} |sim_{a,k}|}$$

where $R_{x,y}$ denotes the rating of user x on item y, R_z' denotes the average rating of user z, and $sim_{v,u}$ denotes the level of similarity between users v and u.

The Similarity Computation stage of the CF requires comparing the active user with every other user in the system. For a ratings matrix storing the ratings of M users on N items, the computational complexity of the Similarity Computation stage is $O(MN)$. This indicates poor scalability of the Similarity Computation, as the complexity grows linearly with both the number of users and the number of items in the matrix. Many prior works have dealt with decreasing the computational effort required by the CF. In general, it is achieved either by preprocessing the ratings matrix, or by distributing the computationally intensive stages of the CF among multiple machines.

Various pre-processing techniques for decreasing the computational effort required by the CF (e.g., correlation coefficients, vector-based similarity, and statistical Bayesian methods) are discussed and analyzed in [3]. Another technique, exploiting pre-clustering of the ratings matrix, is discussed in [6]. There, principal component analysis is used to identify two *discriminative* dimensions of the ratings matrix and all the vectors are projected onto the resulting plane. This inherently partitions the users to clusters or neighborhoods, which are further used to generate the predictions. In [5], the authors use a tree-like data structure and apply a *divide-and-conquer* approach using an iterative K-means clustering to group the users. This leads to smaller and more homogeneous clustering of users for the following Predictions Generation stage.

An alternative approach is to distribute the CF computational effort among the users, such that every user independently computes its similarity with the active user. This approach was initially proposed in [22] and elaborated on in [20]. The latter also developed a detailed taxonomy of the CF distribution approaches and presented implementation frameworks for different application domains. The PocketLens project [11] compared five decentralized distributed architectures for the CF. They showed that the performance of the decentralized mechanism is similar to the performance of the centralized CF while providing increased robustness and security. Further improvements to the decentralized CF were discussed in [8], which proposes the exploitation of Peer-to-Peer platform for a decentralized management of users' profiles. However, this approach approximates the set of the most similar users identified by the Neighborhood Formation stage of the CF, and as a result, the accuracy of the generated predictions is reduced.

This paper is loosely based on the ideas of CAN [15], a content-addressable Peer-to-Peer platform. We implement a fast heuristic variant of the CF, using a CAN-like multi-dimensional space for maintaining a connected structure of users. This allows to significantly decrease the computational effort required by the Similarity Computation and Neighborhood Formation stages by limiting the search process to a search among potentially similar users located in close vicinity to the active user.

3 Content-Addressable Data Management

This section presents the general architecture of CAN [15], a scalable decentralized data management platform. In CAN, the users are represented in a one-to-one manner by the nodes of a virtual *N*-dimensional coordinate space such that the location of the user's node is denoted by a vector $(v_1,v_2,\ldots,v_N)$, where v_i represents the numeric coordinate of the node within a dimension number *i*. In addition to the node, each user continuously manages an *N*-dimensional subspace, called a *zone*. For example, consider a *2*-dimensional space partitioned to *3* zones, managed by users *A*, *B*, and *C* (figure 1-left). Note that the figure shows only the zones managed by the users, whereas the nodes themselves are not shown.

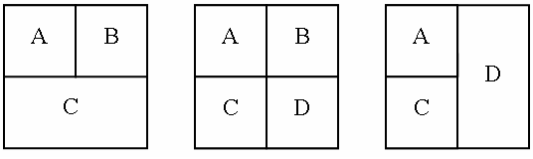


Fig. 1. Example of a *2*-Dimensional CAN space

In CAN space, two nodes (and zones) are called *neighbors* if their coordinate spans overlap along *N-1* dimensions and adjoin along one dimension. For example, consider the neighbor zones *A* and *C* in figure 1-left, whose coordinates partially overlap across the horizontal dimension and adjoin along the vertical. To maintain connectivity in CAN space, each node stores a list of pointers to a set of other nodes, managing the neighbor zones. For example, node *A* stores pointers to the nodes managing zones *B* and *C* (as, respectively, horizontal and vertical neighbors) in its list of pointers.

Routing of messages in CAN space is based on the Plaxton routing algorithm [14]. This routing iteratively forwards the messages to the nodes that are closer to the target node than the current node using a greedy forwarding. The metric for evaluating the distance between two nodes in the address space is the L_1 metric, i.e., the Manhattan Distance. This metric was chosen due to the fact that CAN space inherently supports it, as every node stores a list of pointers to the nodes, managing the neighbor zones. For example, the distance between the nodes (*1,2,3*) and (*6,5,4*) in *3*-dimensional CAN space is *(6-1)+(5-2)+(4-3)=9*. Thus, in *N*-dimensional CAN space a message is routed between an arbitrary pair of nodes in *O(N)* routing steps.

In addition, CAN provides a connectivity maintenance algorithm, stable to sporadic joins and departures of new users. When a new user is inserted, it is assigned its own node and the respective zone. This is done by splitting a zone (determined by the

content provided by the recently inserted user) of one of the existing neighbors according to the following steps: (1) the new user identifies an existing network node, (2) the new user is routed to the target zone that will be split, and (3) the target zone is split and the neighbors of the new zone are updated to maintain connectivity and facilitate future routings. As a result, only a subset of immediate neighbor zones of the zone that was split is actually affected by the insertion of a new node.

The issue of splitting the target zone (i.e., how to split the existing zone, where the contents of the recently inserted node are mapped?) is one of the important issues affecting the performance of CAN. A number of splitting policies are proposed, analyzed and compared in [15]. The simplest policy for the zones splitting is so-called *ordered* splitting. According to this policy, the number of dimension, across which a zone is split, iteratively increases from *1* to *N*. For example, consider user *D* joining CAN *2*-dimensional space (figure 1-middle). Assuming that the content provided by *D* should be located in the right part of the zone managed by node *C* and this is the zone that will be split, user *D* is routed to *C* using the Plaxton routing, and zone *C* is split across the horizontal dimension (assuming that the previous split of *C* was, and the following split of both *C* and *D* will be performed across the vertical dimension). Finally, the recently inserted node, managing the zone *D* notifies its neighbors (i.e., the users managing zones *B* and *C*) about the insertion of a new node, and also their neighbors' lists are updated. Note that only the zone managed by user *C*, which was split and a subset of its neighbor zones (actually, only one zone managed by user *B*), are affected by the insertion of a new user *D*, whereas other zones are not affected.

Disconnections of the users are handled in a similar manner. The disconnecting user identifies one of the neighbor nodes that will takeover managing its zone, and updates other neighbor zones about the departure and the management takeover. For example, consider the user managing zone *B* disconnecting from CAN space (figure 1-right). As a result of the disconnection, the user managing zone *D* takeovers the management the zone previously managed by user *B*.

Thus, CAN provides a decentralized platform, supporting (1) dynamic space partitioning and zones allocation, (2) efficient routing algorithm, and (3) connectivity maintenance algorithm over virtual *N*-dimensional coordinate space. Note that the distributed structure of CAN is not robust against sudden departures of users, as fault-tolerance is not one of the main goals of the platform. However, CAN facilitates a decentralized self-manageable platform for content-addressable data management in a distributed environment.

4 CF over Content-Addressable Space

This work proposes a heuristic variant of the CF. It uses a content-addressable architecture for the purposes of optimizing traditional exhaustive *K*-Nearest Neighbors (KNN) search to a search among potentially similar users only. Although our algorithm is a heuristic one by nature, experimental results demonstrate that it facilitates efficient search process without hampering the accuracy of the generated predictions.

4.1 Mapping User Profiles to Content-Addressable Space

The input for the CF algorithm is a matrix of users' ratings on items, where each row (ratings vector) represents the ratings of a single user and each column represents the ratings on a single item. The total number of items (N) defines an N-dimensional space, where the coordinates range in each dimension corresponds to the range of ratings on the respective item. To handle the ratings matrix in a content-addressable manner, we map it to a CAN-like multi-dimensional space. Each rating is projected using a uniform *injective (one-to-one) mapping* onto the appropriate dimension, such that the whole vector of length N is mapped to a single point in an N-dimensional space. For example, consider a system storing the ratings of users on three different items. In such a system, the evolving CAN-like space will be a *3*-dimensional cube, where the range of coordinates within every dimension corresponds to the range of possible ratings on the respective item.

As already mentioned, each user is represented in a CAN-like space by a single node whose location corresponds to the set of user's ratings and by the respective zone For example, consider a user U that rated all three items in the above *3*-dimensional cube: item i_1 was rated as r_1, item i_2 as r_2, and i_3 as r_3. U will be mapped to a location (r_1,r_2,r_3) of the space and will have exactly two neighbors in each dimension. For example, in the dimension corresponding to item i_1, U will have two neighbors, $N_1=(r_1-x,r_2,r_3)$ and $N_2=(r_1+y,r_2,r_3)$, such that both N_1 and N_2 rated i_2 as r_2 and i_3 as r_3, N_1 rated i_1 below r_1, and N_2 rated it above r_1, and there is no other user that rated i_1 as r', where $r_1-x<r'<r_1$ or $r_1<r'<r_1+y$. Similarly, U will have two neighbors in the dimension corresponding to item i_2 and to item i_3. If there is no user that provided the required combination of ratings, CAN space will maintain connectivity by connecting user U to a further node, which will serve as its virtual immediate.

Note that in the evolving CAN space, the users can be dynamically inserted and removed not only during the initialization, but also during the life cycle of the system. This is explained by the observation that the above connectivity maintenance algorithm guarantees that the structure remains connected regardless of the sudden joins and disconnections of the nodes. Nevertheless, CAN spaces can barely manage insertions of new items, as the dimension of the space should remain fixed. Thus, the proposed heuristic search (that will be discussed in the following sub-section) is applicable only over a stable matrix of users' ratings, where no new items are inserted.

Deciding on the zones split policy affects the evolving structure of the ratings vectors. In our implementation, we used the above mentioned ordered splitting policy. This policy may be sub-optimal in terms of the number of neighbor zones, resulting in a less efficient algorithm, i.e., more comparisons or retrieving less similar neighbors. However, our experiments demonstrate that even this simple policy considerably increases the efficiency of the proposed K-Nearest Neighbors (KNN) search, in comparison with the traditional exhaustive search. Evaluating other splitting policies is beyond the scope of this work.

In addition to the guaranteed connectivity, content-addressable space *inherently clusters* similar users, such that the distance between two similar users (in our case, according to the MSD similarity metric) is lower than the distance between two arbitrary users. This is achieved due to the use of an injective mapping of the ratings vector to the multi-dimensional CAN-like space, which preserves the users' similarity

while mapping the ratings vectors to the numeric coordinates in the space. The following subsection shows a use of the above inherent clustering property for the purposes of developing fast heuristic variant of the KNN search.

4.2 Heuristic Nearest-Neighbors Search

The Neighborhood Formation stage of the CF over the evolving *N*-dimensional space can be schematically described as a heuristically expanding breadth-first search. The algorithm for retrieving *K*-Nearest Neighbors of a user *x* is briefly explained by the following pseudo-code. The code uses two lists of size *K*: (1) *CANDIDATES* – list of candidates for being one of the *K*-nearest neighbors, and (2) *NEIGHBORS* – list of real *K*-nearest neighbors. In principle, the algorithm needs the *CANDIDATES* list only, as the *NEIGHBORS* list only increases during the execution of the algorithm until it reaches its maximal length and contains the real *K*-nearest neighbors. For the sake of clarity, we show an algorithm that uses two lists instead of only one.

```
K_Nearest_Neighbors (user x)
(1)  let NEIGHBORS and CANDIDATES be empty lists, each of size K
(2)  let Z be the zone, where x would be mapped in the CAN space
(3)  foreach u∈ (Zћ neighbors(Z))
(4)          compute distance(x,u)
(5)          insert u into CANDIDATES, s.t. CANDIDATES is sorted
             according to the values of distances(x,u)
(6)  for i=1 to K
(7)          choose v with smallest distance(x,v) from CANDIDATES
(8)          for each w∈ neighbors(v) with unknown distance(x,w)
(9)                  compute distance(x,w)
(10)                 insert w into CANDIDATES, s.t. it remains sorted
                     according  to distance(x,v)
(11)         move v from CANDIDATES to NEIGHBORS
(12) return NEIGHBORS
```

Initially, the algorithm pretends to map the active user *x* to its location in the *N*-dimensional space (step *2*). Next, the algorithm identifies the zone *x* is mapped to, and its neighbors, i.e., users managing the neighbor zones (step *3*). For each of these zones, the degree of similarity, i.e., the distance between *x* and the relevant user, is computed (step *4*). Then, the neighbor users are inserted into the *CANDIDATES* list such that the whole list of candidates users is sorted according to the distances of the users from the active user *x* (steps *4* and *5*). Afterwards, the algorithm iteratively (1) selects *v*, the nearest neighbor stored in the *CANDIDATES* list (step *7*), (2) identifies the neighbors of *v* that are not in the *CANDIDATES* list yet, computes their distances from *x*, and inserts them into the *CANDIDATES*, while keeping the list sorted (steps *8*, *9*, and *10*), and (3) removes *v* from the *CANDIDATES* list and inserts it into the *NEIGHBORS* list. Finally, the algorithm returns the *NEIGHBORS* list (step *12*).

Consider an example execution of the KNN search as illustrated in figure 2. The initial structure of *2*-dimensional space is depicted in figure 2a. Nine users, from *a* to *i*, are inserted into the space and manage the respective zones. Note that also this figure shows only the zones managed by the users, whereas the nodes representing the users are not shown. Assume that the active user is mapped to the zone managed by user *e*. Thus, *e* and its neighbors, i.e., users managing zones *c*, *d*, *f*, and *i*, are the first candidates for being the nearest neighbors and they are inserted into the *CANDIDATES* list. Assume

that the user managing zone *e* is the closest one. It is moved from the *CANDIDATES* list to the *NEIGHBORS* list (figure 2a). Since all the neighbors of *e* are already known, the next closest neighbor is chosen among its neighbors. Assume that the next closest neighbor is the user managing zone *f*. It is moved from the *CANDIDATES* list to the *NEIGHBORS* list, and its only new neighbor, the user managing zone *g*, is inserted into the *CANDIDATES* list (figure 2b). The next closest neighbor is the user managing zone *c*, inserting the user managing zone *b* into the *CANDIDATES* list (figure 2c). Assume that the next closest neighbor is the user managing zone *g* (not an immediate neighbor of *e*). As a result, the user managing zone *h* is inserted into the *CANDIDATES* list (figure 2d). This process is repeated until the *NEIGHBORS* list contains *K*-Nearest Neighbors.

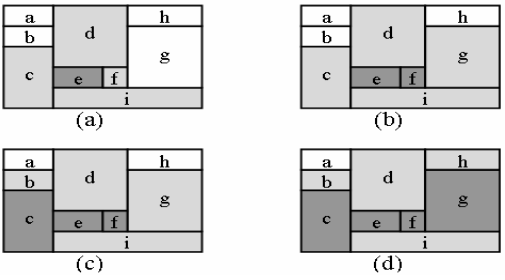


Fig. 2. Stages of the KNN search over *2*-Dimensional CAN space (zones managed by users from the CANDIDATES are in a light and from the NEIGHBORS – in a dark gray)

The proposed algorithm reduces the computational effort required by the Similarity Computation and the Neighborhood Formation stages, in comparison with the traditional CF algorithm, where an active user is compared with all the available users. Conversely, the proposed heuristic algorithm compares the active users with potentially similar users only, located in close vicinity to the active user. Since every user in the *N*-dimensional space continuously maintains an updated list of its immediate neighbors, any neighbor of a given user is accessed through a single network hop. This is true regardless of the physical (geographical) and logical (similarity) distances between the neighbors. Thus, the algorithm will also work in sparse spaces, where the distance between neighbors in the underlying network might be very high.

4.3 Heuristic Completions of User Profiles

In the former sections, we assumed that the user's ratings were represented as a *complete vector*, i.e., explicit ratings on all the items are available. Thus the mapping of the user's ratings vectors to the underlying content-addressable space is straightforward. However, this assumption is unachievable in most real-life applications and scenarios, where an average user rates only a portion of the available items. This raises a need for developing a mapping mechanism capable of mapping *incomplete vectors*, where a subset of the ratings is missing, to the content-addressable space. In this sub-section we propose three mappings to handle this task. However, instead of developing a new mapping of incomplete vectors to the content-addressable space, we propose to convert the incomplete vectors to complete ones by heuristically filling-in

the missing ratings in the incomplete vectors [2]. Thus, the proposed completion heuristics are designed to re-use the above injective mapping of complete vectors, while employing it on the modified vectors with heuristically filled-in ratings.

As the completion heuristics are not the main focus of the current work, we suffice with three relatively simple heuristics that demonstrate the applicability of the proposed vectors' completion. The heuristics are as follows:

- *User-average* – the missing rating on an item in the user's vector is substituted with the average of the real ratings, explicitly provided by this user.
- *Item-average* – the missing rating on an item in the user's vector is substituted with the average of the real ratings, explicitly provided by the other users on this item.
- *Conditional* – integrates both the user-average and the item-average heuristics and decides in a run-time regarding the specific completion heuristic to be used according to a certain predefined condition.

Clearly, the *user-average* heuristic can be considered as an accurate personalized completion heuristic, as the missing ratings are substituted with a value, produced by the real ratings of the given user. Thus, it reflects the real preferences and tendencies of the user, such as over- or under-rating of items, natural intensity of expressions and so forth. Conversely, the *item-average* heuristic can be considered as the most accurate non-personalized completion heuristic, as the missing ratings are substituted with a value, produced by numerous real ratings on the given item. As such, it reflects a general (and relatively reliable) opinion of many other users on the item. We conjecture that the *user-average* heuristic is preferable when the knowledge about the user's preferences is reliable, i.e., the number of ratings explicitly provided by the user is relatively high. On the other hand, when the number of user's explicit ratings is low, the *item-average* heuristic will exploit other users' ratings for filling-in the missing rating and it should be preferred. Based on these considerations, we defined another *conditional* heuristic, which will autonomously decide which of the above completion heuristics should be exploited for filling-in the missing ratings of every user.

In summary, each of these heuristics allows the filling-in of the missing ratings, converting the incomplete vectors to the complete ones, and then mapping them to the content-addressable space using the above mentioned injective mapping mechanism.

5 Experimental Evaluation

In the experimental part of our work we used the Jester dataset of jokes' ratings [6]. Jester is a Web-based joke Recommender System, containing *4.1* millions of ratings (on a continuous scale from *–10.00* to *+10.00*) of *73,421* users on *100* jokes. A significant portion of the users rated all the jokes, so the Jester dataset is relatively dense. Overall, approximately *56%* of all the possible ratings in the matrix are present. For the complete vectors experiments, we selected a subset of *14,192* users that rated all *100* jokes, producing a matrix, where every value corresponds to a real rating, explicitly provided by a user. The average rating of a joke is *0.807*, and the standard deviation of the ratings in the matrix is *4.267*. We implemented a centralized simulation of a *100*-dimensional CAN space and inserted the above *14,192* users into the space. Insertions of the users into the space were done using the ordered splitting policy.

5.1 Scalability of the Search

These experiments were designed to evaluate the scalability of the proposed heuristic variant of the KNN search. The efficiency of CAN-based KNN is measured by the number of comparisons performed by the Neighborhood Formation stage of the CF.

In this experiment we measured number of comparisons during the Neighborhood Formation stage. For this, we gradually increased the number of users inserted into the system from *M=1,000* to *M=14,000*. For each *M*, we computed the number of comparisons performed in the traditional exhaustive KNN search and in CAN-based heuristic variant of KNN. Both searches were aimed at retrieving *K=5* nearest neighbors. For each value of *M*, the experiments were repeated *1,000* times for different active users. The results are shown on Figure 3. The horizontal axis stands for *M*, the number of users inserted into the system, and the vertical axis reflects the average number of comparisons during a single KNN search, for both exhaustive and heuristic searches.

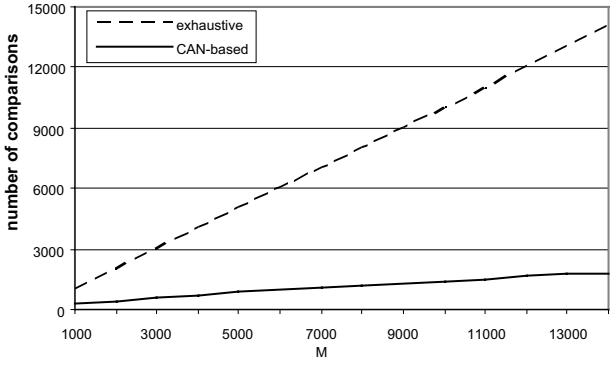


Fig. 3. Average number of comparisons vs. the number of users inserted

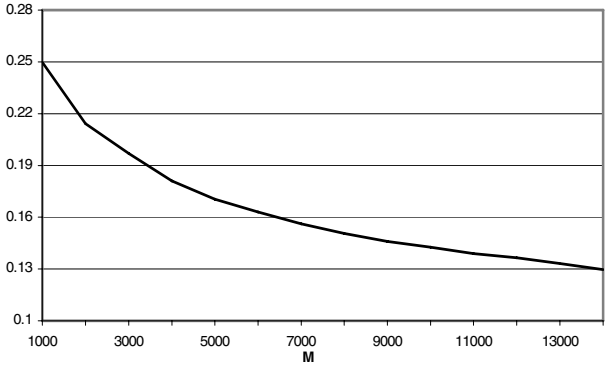


Fig. 4. Ratio between the number of comparisons vs. the number of users inserted

As expected, the number of comparisons in CAN-based KNN is significantly lower than in traditional KNN and it grows at a logarithmic-like manner with the number of users. This is explained by the fact that in CAN-based KNN the active user is compared only with a subset of highly similar users (located in close vicinity in a content-addressable space), whereas in traditional KNN it is exhaustively compared with all the available users. To better understand the scalability of the proposed approach, we computed the ratio between the number of comparisons in CAN-based KNN and the number of comparisons in the exhaustive KNN. This ratio was computed for different values of *M* and the results are shown on Figure 4. It can be seen that the ratio steadily decreases with *M*. This allows us to conclude that the proposed algorithm is applicable in large-scale systems with high number of users and items, e.g., on the Web.

The second experiment was designed to evaluate the scalability of CAN-based KNN with the number of nearest neighbors (*K*) to be retrieved. We gradually increased the value of *K* from *K=1* to *K=50*. For each value of *K*, we measured the number of comparisons needed to retrieve *K* nearest neighbors for *M=1,000*, *2,000*, *4,000*, *8,000*, and *14,000* users. For each value of *M* and *K*, the experiments were repeated *1,000* times for different active users. The number of comparisons as a function of *K* for the above values of *M* is shown on Figure 5. The horizontal axis stands for *K*, the number of nearest neighbors to be retrieved, whereas the vertical reflects the average number of comparisons during the KNN search.

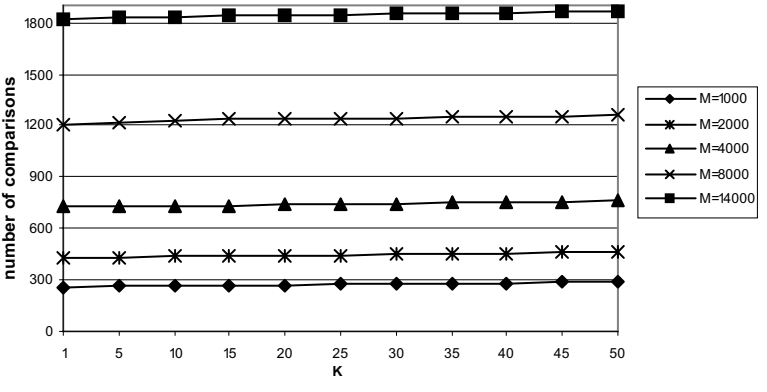


Fig. 5. Average number of comparisons vs. the number of retrieved neighbors

As can be seen, the number of comparisons in CAN-based KNN remains roughly unchanged when *K* increases. This is explained by the observation that most of the KNN users are located in close vicinity to the active user. Thus, the similar users are discovered in the early stages of the KNN search, while further expansions contribute very few new similar users. Both experiments show good scalability of CAN-based KNN with *K*. This means, that practical Recommender Systems can use higher values of *K*, to form moderately larger and more reliable neighborhoods, and generate more accurate predictions with only a very minor computational overhead.

5.2 Accuracy of the Search

The following experiments were designed to evaluate the accuracy of the results obtained by the proposed heuristic variant of KNN search. In the first experiment we compared the sets of users, i.e., the neighborhoods, retrieved by the traditional (exhaustive) KNN and by the CAN-based variant of KNN.

Let us denote by KNN_e the set of users retrieved by the traditional exhaustive KNN search and by KNN_h the set of users retrieved by the CAN-based heuristic variant of KNN. Since the CAN-based KNN is a heuristic approach, a sub-optimal structure of zones may lead to a situation, where $KNN_e \neq KNN_h$, i.e., the heuristic search retrieves only a subset of the real K nearest neighbors. As the collaborative predictions are generated by aggregating the ratings of similar users, identifying the set of most similar users is essential for generating accurate predictions.

To evaluate the accuracy of the proposed heuristic KNN search, we adapt the traditional Information Retrieval metric of precision [18]. In fact, the computed accuracy metric is rather *precision@K*, since the overall search procedure is limited to K most similar users only. However, this metric also provides some indication about the recall of the search, as it can be considered as the recall of the search for a limited number of the most similar users to be retrieved. For the sake of clarity, this metric is referred to in the paper as *precision*. The precision is computed by:

$$precision = \frac{|KNN_e \cap KNN_h|}{|KNN_e|} = \frac{|KNN_e \cap KNN_h|}{K}$$

The cardinality of the KNN_e set was $K=10$, while the cardinality of the KNN_h set was gradually increased from $K'=1$ to $K'=100$. The precision was computed for $M=1{,}000$, $2{,}000$, $4{,}000$, $8{,}000$ and $14{,}000$ users inserted into the system. For each value of M and K', the experiments were repeated $1{,}000$ times for different active users. Figure 6 shows the precision as a function of K' for the above values of M. The horizontal axis stands for M, the number of users inserted into the system, whereas the vertical reflects the average precision of the heuristic KNN search.

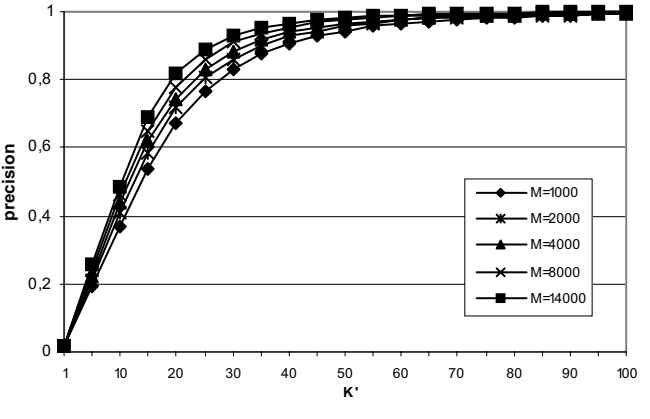


Fig. 6. Precision of CAN-based KNN

As can be seen, the curves behave similarly and the accuracy increases with K', such that for $K'>50$ it is over *0.9* for all the given values of M. Previous experiments presented in previous sub-section show that the algorithm is highly scalable with K. Thus, retrieving a larger set of users (i.e., higher values of K') leads to a minor increase in the computational overhead. Hence, it is feasible to moderately increase the number of neighbors retrieved by CAN-based search in order to achieve a higher accuracy and generate better predictions.

Since the precision of the heuristic CAN-based KNN search may seem low for small values of K', we conducted another two experiments, aimed at evaluating the quality of the neighborhood retrieved by the heuristic search. In the first, this was done by computing the average similarity between the nearest neighbors retrieved by the heuristic search and the active user. The computed average similarity was compared to the average similarity of neighborhood retrieved by the traditional search.

In the experiment, we gradually increased the number of users inserted into the system from $M=1,000$ to $M=14,000$. For each value of M, we compared the average similarity of heuristically retrieved neighbors with the average similarity of exhaustively retrieved neighbors for $K=K'=10$. For each value of M, the above experiments were repeated *1,000* times for different active users. The results of the experiment are shown on Figure 7 (discussed after Figure 8). The horizontal axis stands for the number of users inserted into the system, whereas the vertical reflects the average similarity between the users in the *KNN* set and the active user for both searches.

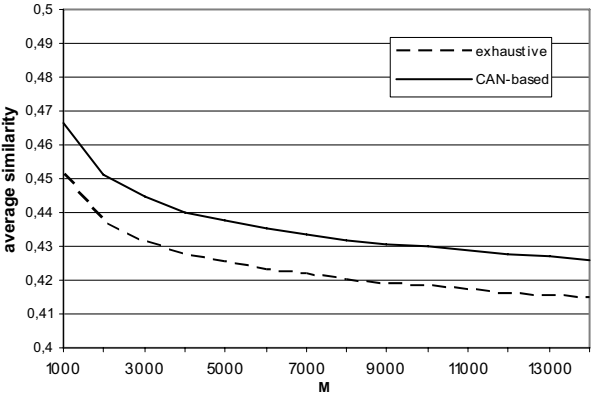


Fig. 7. Average similarity vs. the number of users inserted

The second experiment was designed to evaluate the quality of the heuristically retrieved neighborhood by comparing the accuracy of the generated predictions. The final goal of the KNN search is to retrieve a set of the most similar users, whose ratings will be aggregated when generating the predictions. Thus, we generated the predictions using both exhaustively and heuristically retrieved sets of K-Nearest Neighbors and evaluated the accuracy of the predictions using well-known Mean Average Error (MAE) metric [9]:

$$MAE = \frac{\sum_{i=1}^{N} | p_i - r_i |}{N}$$

where N is the number of predictions, p_i and r_i are the predicted and real ratings i.

Also in this experiment the number of users inserted into the system was gradually increased from *M=1,000* to *M=14,000*. For each value of *M*, the experiment was repeated *1,000* times for various, randomly chosen active users. For each active user chosen, the following operations were conducted: (1) a single randomly selected rating in the user's profile was hidden and served as a rating to be predicted, while the remaining *all-but-one* ratings served as the user's profile, (2) basing on the all-but-one user's profile, the set of *K=K'=10* nearest neighbors was retrieved using both traditional exhaustive and heuristic retrievals, (3) predictions were generated using both heuristically and exhaustively retrieved neighborhoods, and (4) the MAE error of the generated predictions relatively to the original hidden rating was computed. The average values of the MAE computed for certain values of *M* are shown on Figure 8. The horizontal axis stands for the number of users inserted into the system, whereas the vertical reflects the MAE values for both exhaustive and heuristic searches.

The results show that the average similarity (which is actually the dissimilarity) and the MAE of the predictions decrease with *M*. This is explained by the observation that the probability of discovering a similar user increases with the number of users inserted into the system. Thus, the average dissimilarity of the retrieved *K*-Nearest Neighbors decreases with *M*, while the accuracy of the generated predictions increases, and the MAE decreases as well.

Although both the similarity and the MAE of CAN-based heuristic search are higher (i.e., the retrieved neighbors are more dissimilar and the accuracy is actually lower), the curves are very close and the results are quite similar. Average deviation of the similarities is *2.93%* and of the MAEs is only *0.38%*. Note that the average deviation of the MAE is significantly lower than the average deviation of the similarities, as the generated predictions are barely affected by the changes in the retrieved neighborhoods. These experiments allow us to conclude that the proposed heuristic algorithm succeeds in both retrieving similar neighborhoods and generating accurate predictions.

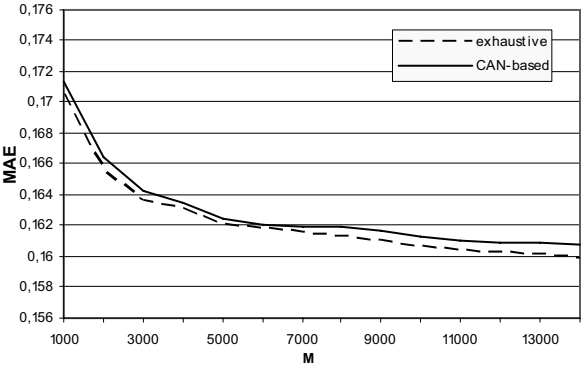


Fig. 8. Mean Average Error of the predictions vs. the number of users inserted

5.3 Inherent Clustering

One of the basic assumptions, that allows us to limit the heuristic search to users, located in close vicinity to the active user, is the *inherent clustering*. That means that

the distance between two similar users is lower than the distance between two arbitrary users. Thus, the following experiment was designed to verify the property of inherent clustering in the underlying content-addressable space.

For this, we computed the average and the standard deviation of the similarity of the users located *R=1*, *2*, and *3* routing hops from the active user. The experiments were conducted for *M=1,000*, *2,000*, *4,000*, *8,000* and *14,000* users inserted into the system. For each value of *M*, the experiments were repeated *1,000* times for different orders of inserting the users into the system and for different active users. Figure 9 shows the average similarity and the standard deviation as a function of *R* for the above values of *M*. The horizontal axis stands for *M*, the number of users inserted into the system, whereas the vertical reflects the average and the standard deviation of the similarity of the retrieved users within a given number of hops from the active user.

It can be seen that for any value of *M* the similarity increases with *R*. This means that the similarity of users, located close to the active user is higher than the similarity of those located far. Thus, this experiment verifies our assumption on the clustering in content-addressable space. For any *R*, the average similarity and the standard deviation steadily decrease with *M*. This is explained by the fact that higher number of users leads to a better organization of zones, where zones managed by more similar users *block* the zones managed by dissimilar users. Thus, the average similarity (and the standard deviation) of users located within a given number of hops decreases with *R*.

Moreover, this experiment demonstrates the stability of the proposed CAN-based structure of users. This experiment was repeated *1,000* times, for different random orders of inserting the users into the system. Low values of the standard deviation, and the steady decrease of it with the number of users in the system, show that the inherent clustering holds regardless of the different types of organization of the CAN zones, imposed by the different orders of inserting the users. Thus, the proposed KNN search will succeed to retrieve accurate neighborhoods for various system usage scenarios.

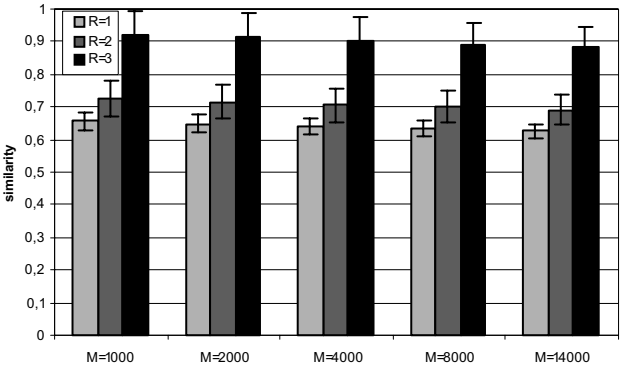


Fig. 9. Average similarity vs. number of hops from the active user

5.4 Completion Heuristics

The following experiments were designed to evaluate the proposed completion heuristics for filling-in the missing values in the incomplete ratings vectors. To run the experiment with incomplete vectors, we used the full Jester dataset [6]. In previous

experiments we used a partial dataset of complete vectors, built by *14,192* users that rated all *100* jokes. In addition, the full dataset also contains the ratings of *59,229* users that rated on average *45.26* jokes. The full Jester dataset (containing both complete and incomplete vectors) was used in the completion heuristics experiments.

We implemented the *user-average* and the *item-average* completion heuristics. As for the *conditional* heuristic, the decision regarding the chosen completion heuristic was based on the number of rated items in user's ratings vector. Since in the full Jester dataset the average number of items rated by a user was *45.26*, in our implementation of the *conditional* heuristic the threshold was set to *20* items. This means that if a user rated less than *20* items, her ratings vector is not considered reliable, and the *item-average* heuristic is applied. Otherwise, the *user-average* heuristic is applied.

To evaluate the accuracy of the proposed completion heuristics, we conducted two experiments. In the first, we compared the average similarity between the active user and the *K*-Nearest Neighbors retrieved by the heuristic search and by the traditional exhaustive search. The experiment was repeated three times, for the different completion heuristics exploited before inserting the completed vectors to the underlying content-addressable space. In the experiment, we gradually increased the number of users inserted into the system from *M=5,000* to *M=50,000*. For each value of *M*, we compared the average similarity of the retrieved neighbors (using both exhaustive and heuristic retrieval techniques) for *K=K'=10*. For each value of *M*, the experiments were repeated *1,000* times for different active users. The results of the experiment are shown on Figure 10. The horizontal axis stands for *M*, the number of users inserted into the system, whereas the vertical reflects the average similarity between the users in *KNN* set and the active user for exhaustive and heuristic searches.

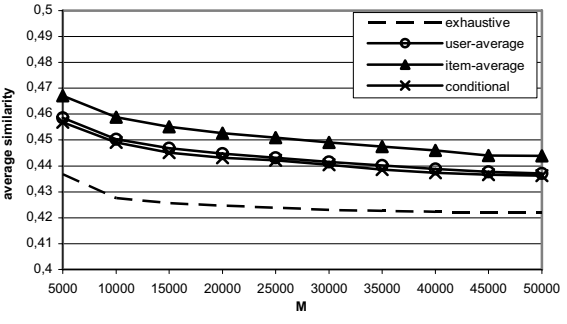


Fig. 10. Average similarity vs. the number of users inserted

The curves show, that similarly to the accuracy results in previous sub-sections, the average similarity (i.e., dissimilarity) of the retrieved KNN users decreases with *M*, the number of users inserted into the system. Comparison of the proposed completion heuristics yields that the personalized *user-average* heuristic outperforms the non-personalized *item-average* heuristic. Average similarity deviation of the KNN set exploiting the *user-average* heuristic from the exhaustively retrieved KNN is *4.43%*, while the similarity deviation of the *item-average* KNN set is *6.21%*. Since the *conditional* heuristic is a smarter combination of the above heuristics, it slightly outperforms

the *user-average* heuristic as well, and for it the average similarity deviation from the exhaustively retrieved KNN set is *4.11%*.

Since the goal of the Collaborative Filtering is to generate predictions, the second experiment was designed to evaluate the quality of the completion heuristics by comparing the accuracy of the generated predictions. Hence, we generated the predictions using both exhaustively and heuristically retrieved sets of *K*-Nearest Neighbors and evaluated the accuracy of the predictions using the MAE metric. In the experiment the number of users inserted into the system was gradually increased from *M=5,000* to *M=50,000*. For each value of *M*, the experiment was repeated *1,000* times for various active users.

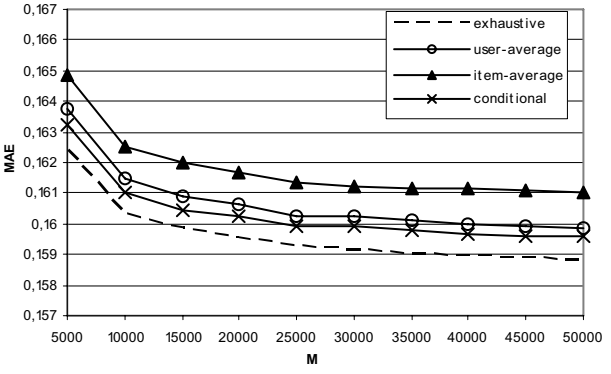


Fig. 11. Mean Average Error of the predictions vs. the number of users inserted

The experimental setting was similar to the previous sub-sections: the sets of *K=K'=10* nearest neighbors was retrieved using both exhaustive and heuristic retrievals, the predictions were generated using both neighborhoods, and the MAE of the generated predictions relatively to the original rating was computed. The average values of the MAE are shown on Figure 11. The horizontal axis stands for the number of users inserted into the system, while the vertical reflects the MAE values for both exhaustive and heuristic searches. Note that the heuristic retrieval was conducted three times, according to the completion heuristics being exploited.

Similarly to previous results, this experiment shows that the MAE of the prediction decreases with *M*, the number of users inserted into the system. Comparison of the proposed completion heuristics yields that the accuracy of the predictions using personalized *user-average* heuristic is better than of the non-personalized *item-average* heuristic. However, for both heuristics the increase of the MAE is minor: for the *user-average* heuristic it is *0.69%*, whereas for the *item-average* heuristic it is *1.37%*. As can be seen, also in this experiment the *conditional* heuristic outperforms both of them, as for the *conditional* heuristic the increase of the MAE is *0.46%*. Hence, out of the proposed completion heuristics, the *conditional* heuristic retrieves the most similar KNN set and generates the most accurate prediction. This allows us to conclude that this heuristic should be used for converting the incomplete vectors to complete ones, and leads to future research on developing more accurate completion heuristics.

6 Conclusions and Future Work

One of the major drawbacks of the state-of-the-art CF implementations is their high computational complexity, which grows linearly both with the number of users and items in the system. In this work we proposed to heuristically decrease the required computational effort by implementing the CF over content-addressable CAN-like *N*-dimensional space.

Experiments conducted over the Jester dataset of joke ratings show that the proposed heuristic algorithm outperforms the traditional exhaustive KNN search as the computational overheads are significantly decreased, while the accuracy remains similar. Our algorithm decreases the number of required comparisons, such that the ratio between the numbers of comparisons steadily decreases with the number of users. For example, for *14,000* users the number of comparisons was decreased by *87%*. Other experiment shows that the number of comparisons roughly remains unchanged when *K* increases. This allows us to increase the number of nearest neighbors retrieved (and to improve the accuracy of the predictions) with a minor computational overhead.

In the accuracy experiments we qualitatively compared the neighborhoods retrieved and the predictions generated by the CAN-based heuristic and by the traditional exhaustive KNN searches. The retrieved neighborhoods were similar and the predictions were close, which indicates a good accuracy of the proposed algorithm. In summary, comparing the proposed heuristic search with traditional exhaustive search shows that our algorithm achieves high accuracy, while significantly decreasing the required computational effort. Another experiment was aimed at validating the inherent clustering property of content-addressable spaces. The results showed that this property holds in the CAN-like space, as the dissimilarity of users, located in a certain number of network hops from the active user increased with the number of network hops. The experiments also showed that the inherent clustering property holds regardless of the number of users inserted into the system and the order of their insertion.

Finally, the last experiment was aimed at comparing three simple heuristic for converting the incomplete vectors to complete ones by filling-in the missing ratings. The experimental results showed that the heuristic, which conditionally integrates two other heuristics, outperforms them both in terms of the retrieved neighborhoods' similarity and of the generated predictions' accuracy. Comparing the MAE of the predictions generated by the complete and heuristically completed vectors yields that the accuracy of the predictions generated by the complete vectors is slightly better. This conclusion is reasonable, since the proposed completion heuristics insert some extent of noise into the original ratings. However, the increase in the MAE is minor, allowing us to conclude that the achieved computational optimization is preferential than the minor noises in the predictions caused by the artificial ratings inserted by the completion heuristics.

In this work, we inherently assumed that the system assigns equal relative weights to the ratings on each item. However, this assumption is not true in many real-life personalization applications. For example, this assumption might be false in a situation, where different criteria affect differently on the similarity values, e.g., when the similarity values between the items are known. Developing a weighted prediction algorithm will result in a more accurate Recommender System.

Also, we assumed that either the user's ratings on the items are available or they can be easily filled-in using one of the proposed simple completion heuristics. However, some real-life scenarios, this completion is hard to achieve, since the matrix is very sparse (e.g., density of *2-3%* in typical Collaborative Filtering datasets such as [9] and [10]) and the substitution of the missing values may require exploiting more intelligent techniques. In the future, we plan to study the use of various completion heuristics, exploiting statistical and Machine Learning techniques.

In addition to decreasing the computational effort, the proposed algorithm can naturally be extended to distribute it among multiple users. In traditional centralized implementations of the CF, the Similarity Computation and the Neighborhood Formation stages are performed in a single central location. However, as the underlying CAN platform is originally distributed Peer-to-Peer platform, it inherently allows distributed and fully decentralized storage of the ratings matrix. In the future, we plan to implement a distributed variant of the algorithm and to investigate the distribution issues.

The current work is limited to the Mean Squared Difference (MSD) similarity metric, since the injective mapping to a multi-dimensional CAN-like space inherently supports it. However, for other metrics, such as Cosine Similarity or Pearson Correlation, CAN space might be inappropriate and new types of topologies and respective mappings should be developed. We plan to study other metrics and to produce a general framework for efficient heuristic Collaborative Filtering.

References

1. Aguzzoli, S., Avesani, P., Massa, P.: Collaborative Case-Based Recommender System. In: Proceedings of the ECCBR Conference (1997)
2. Bogaerts, S., Leake, D.: Facilitating CBR for Incompletely-Described Cases: Distance Metrics for Partial Problem Descriptions. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, Springer, Heidelberg (2004)
3. Breese, J., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the UAI Conference (1998)
4. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
5. Chee, S.H.S., Han, J., Wang, K.: RecTree: An Efficient Collaborative Filtering Method. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2001. LNCS, vol. 2114, Springer, Heidelberg (2001)
6. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A Constant Time Collaborative Filtering Algorithm. Information Retrieval Journal 4(2) (2001)
7. Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining Collaborative Filtering with Personal Agents for Better Recommendations. In: Proceedings of the AAAI Conference (1999)
8. Han, P., Xie, B., Yang, F., Shen, R.: A Scalable P2P Recommender System Based on Distributed Collaborative Filtering. Expert Systems with Applications Journal 27(2) (2004)
9. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of the SIGIR Conference (1999)
10. McJones, P.: Eachmovie Collaborative Filtering Data Set (1997), http://research.compaq.com/SRC/eachmovie/

11. Miller, B.N., Konstan, J.A., Riedl, J.: PocketLens: Toward a Personal Recommender System. ACM Transactions on Information Systems 22(3) (2004)
12. Morita, M., Shinoda, Y.: Information Filtering Based on User Behavior Analysis and Best Match Retrieval. In: Proceedings of the SIGIR Conference (1994)
13. Pennock, D.M., Horvitz, E., Giles, C.L.: Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering. In: Proceedings of the AAAI Conference (2000)
14. Plaxton, C., Rajaraman, R., Richa, A.: Accessing Nearby Copies of Replicated Objects in a Distributed Environment. In: Proceedings of the ACM SPAA Conference (1997)
15. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content-Addressable Network. In: Proceedings of the SIGCOMM Conference (2001)
16. Resnick, P., Varian, H.R.: Recommender Systems. Communications of the ACM 40(3) (1997)
17. Ricci, F., Venturini, A., Cavada, D., Mirzadeh, N., Blaas, D., Nones, M.: Product Recommendation with Interactive Query Management and Twofold Similarity. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS, vol. 2689, Springer, Heidelberg (2003)
18. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Publishers, New York (1983)
19. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of Recommendation Algorithms for E-Commerce. In: Proceedings of the EC Conference (2000)
20. Sarwar, B.M., Konstan, J.A., Riedl, J.: Distributed Recommender Systems: New Opportunities for Internet Commerce. In: Internet Commerce and Software Agents: Cases, Technologies and Opportunities, Idea Group Publishers, USA (2001)
21. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating "Word of Mouth". In: Proceedings of the CHI Conference (1995)
22. Tveit, A.: Peer-to-Peer Based Recommendations for Mobile Commerce. In: Proceedings of the WMC Workshop (2001)

Ontology-Driven Knowledge Sharing for Networked Organisation Configuration

Alexander Smirnov, Tatiana Levashov, and Nikolay Shilov

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), 14 Line VO, 19178, St. Petersburg, Russia
{smir,oleg,nick}@iias.spb.su

Abstract. Distributed networks of independent companies (networked organisations) are currently of high interest. Configuration of a networked organisation is a strategic task that requires intelligent decision support and integration of various tasks constituting the configuration problem. Achieving efficient integration of tasks is possible when it is done taking into account semantics. The paper proposes an approach to this problem based on ontology-driven knowledge sharing. The knowledge in the approach is presented using formalism of object-oriented constraint networks. Application of the developed approach is demonstrated at two levels: (a) at the level of information integration within one company and (b) at the level of information integration across a networked organisation.

Keywords: Networked organisation, decision support, ontology, knowledge sharing, semantic interoperability.

1 Introduction

Global changes in the economy worldwide have led to changes in priorities and strategies of market players. This has caused appearance of new network-driven organisational forms such as virtual enterprises, extended enterprises, supply chains, etc.

A networked organization [1–3] is usually defined as an organization formed by geographically distributed independent partners on the basis of information technologies. Efficient creation of an effective configuration of the networked organisation can give its members a competitive advantage in getting an order. Hence, configuration of the networked organisation is a problem of the strategic level requiring intelligent decision support.

This problem has been addressed in numerous research efforts. However most of them solve particular tasks of the complex problem. A complex approach is required to provide for integration of the tasks to be solved. Among the tasks the following most important ones can be selected (they do not pretend to be a complete list):

1) Order configuration (configuration of the product / service in accordance with existing constraints and customer preferences);
2) Partner choice among existing companies – potential members of the networked organization;

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 179–193, 2008.

3) Resource allocation among the networked organisation members;
4) Transportation network configuration (this logistics related task is required due to the distributed nature of the networked organisation);
5) Configuration of technological resources of the networked organisation members.

During the process of the networked organisation configuration and management the above tasks have to be solved jointly. Hence, it is reasonable to speak about integration of them. Integration of tasks is more than data integration. It requires integration at the level of semantics or semantic interoperability. In other words, it requires knowledge integration.

Knowledge management has shown its efficient applicability in this area. It is a complex cooperative network-centric process to support multi-object and multi-disciplinary areas including modelling, design, knowledge representation and acquisition, decision support and supporting environment [4]. A number of efforts have been done in the area of sharing information and processes between applications, people and companies. However knowledge sharing / exchange requires more than this. It requires information coordination and repository sharing with regard to semantics.

To address this, the paper proposes usage of the idea of Knowledge Logistics (KL) that stands for acquisition, integration, and transfer of the right knowledge from distributed sources to right persons (decision makers) at the right time for the right business purpose in the right context [5]. KL with regard to individual customer requirements, available knowledge sources, and current situation analysis in an open information environment addresses problems of intelligent support of customer activities.

One of the main issues to resolve is interoperability. It can be defined as the ability of enterprise software and applications to interact. Interoperability is considered to be achieved if the interaction can, at least, take place at three levels: data, application and business enterprise [6]. Semantic interoperability assumes interaction at one more level, namely at the level of semantics. To provide for semantic interoperability KL uses ontologies as one of the most advanced approaches to knowledge mark-up and description. Ontologies establish a joint terminology between members of a community of interest [7]. This makes it possible to provide for semantic interoperability between various tasks of configuration and management.

The paper is structured as follows. Approach description is presented in sec. 2. Sec. 3 outlines the principles of the central integrated ontological model creation. The implementation details including standards and protocols are described in sec. 4. The case study is given in sec. 5. Some results are summarised in conclusions.

2 Approach

The KL problem in the presented approach is considered as a configuration of a network of end-users, knowledge resources, and tools & methods for knowledge processing located in the network-centric environment. Such a network of loosely coupled sources is referred to as a knowledge source network or “KSNet” (detailed description can be found in [8]), and the approach is called KSNet-approach. The approach is built upon constraint satisfaction / propagation technology for problem solving since application of constraint networks allows simplifying the formulation and interpretation of

real-world problems that are usually presented as constraint satisfaction problems in such areas as management, engineering, etc. (e.g., [9]).

Selected tasks are integrated using a single integrated ontological model of the networked organisation. In other words it can be stated that these tasks are formulated using the semantics provided by the ontological model. Fig. 1 represents the integration of the tasks with their input and output parameters. It also shows methods for task solving used for the implementation of the approach. Each task in the figure is represented as a rectangle with three boxes in the bottom: input data (left), task solving method (middle), and output data (right). Customer order is considered as a driver for the entire system and the output is a feasible configuration of the networked organisation.

The proposed ontology-driven architecture deals with ontologies of different types. The ontologies are represented by means of a common notation and a common

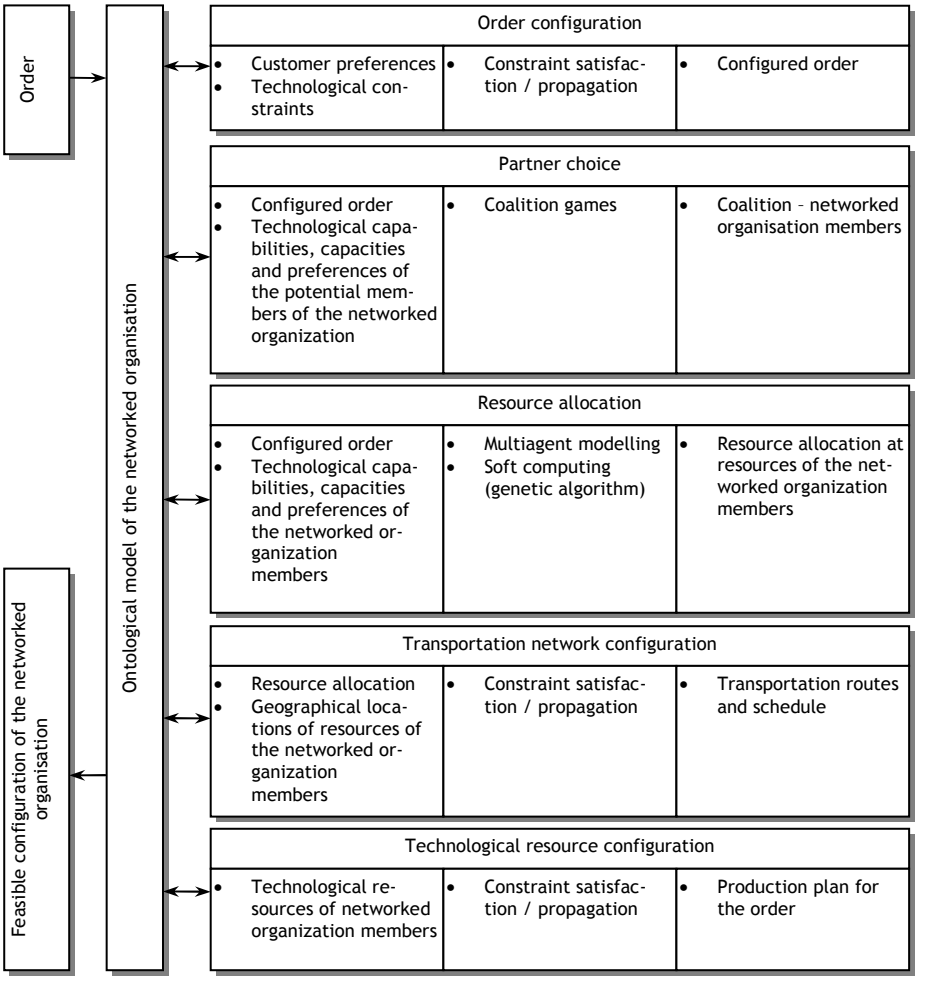


Fig. 1. Tasks solved during configuration of the networked organisation

vocabulary supported by an ontology library. The common representation enables performance of operations on ontology integrations as alignment and merging, and operations on context integrations. Main components of the ontology library are domain, tasks & methods, and application ontologies. All the ontologies are interrelated so that an application ontology is a specialization both of domain and tasks & methods ontologies.

The classification of knowledge according to the abstraction and types [10] distinguishes *universal*, *shared*, *specific*, and *individual* knowledge abstraction levels. In the knowledge sharing model of the system "KSNet" (Fig. 2) the universal level is considered as the common knowledge representation paradigm. The universal level is based on the formalism of object-oriented constraint networks represented by means of a knowledge representation language. The abstractions provided at this level are shared by the ontologies stored in the ontology library. Both shared abstraction level and specific abstraction level are considered sharable and reusable since ontologies of these levels share common representation paradigm and common vocabulary.

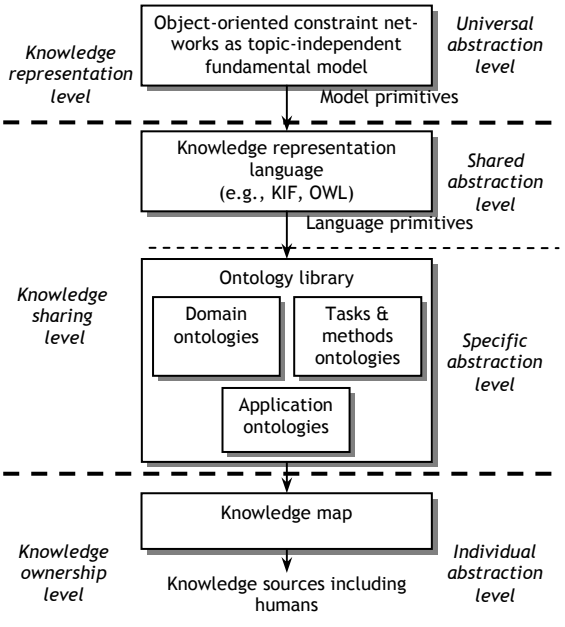


Fig. 2. Ontology-driven knowledge sharing

The level of knowledge representation provides with a common notation for knowledge description and enables compatibility of different formats (e.g., KIF, OWL, etc.). Knowledge sharing level focuses on ontological knowledge common for a particular area. Knowledge represented by this level suits well for sharing and reuse, since this level does not concentrate on any specific properties and, at the same time, knowledge of this level is not a universal abstraction rarely taken into account when the case considers practical knowledge sharing and reuse. The knowledge ownership level increases scalability of the system regarding to the number of knowledge sources that can be attached to the system and users that can be served.

3 Ontological Model

As a general model of ontology representation in the system "KSNet" implementing the approach, the object-oriented constraint network paradigm [5] is used. This model defines the common ontology notation used in the system. In accordance with this representation the ontological model is defined as follows: $M = (O, Q, D, C)$. This formalism includes a set of classes O and attributes Q, Cartesian product of which is a set of variables. Each variable may take values from its domain $D(1),\ldots, D(n),\ldots$.

The model also includes constraints of six types: $C_1, \ldots, C_6 \subset C$, defining which values the variables may take simultaneously, and relationships between classes. To solve a constraint satisfaction task means to assign values to each variable so that all constraints hold. Class "*Thing*" is used as a parent class for all classes of the ontological model, i.e., any class of the ontological model is a direct or indirect child ("is-a" relationship) of the class "*Thing*".

The following types of the constraints have been defined:

$C^I = \{c^I\}$, $c^I = (o, q)$, $o \in O$, $q \in Q$ – accessory of attributes to classes;

$C^{II} = \{c^{II}\}$, $c^{II} = (o, q, d)$, $o \in O$, $q \in Q$, $d \in D$ – accessory of domains to attributes;

$C^{III} = \{c^{III}\}$, $c^{III} = (\{o\}, \text{True} \vee \text{False})$, $|\{o\}| \geq 2$, $o \in O$ – classes compatibility (compatibility structural constraints);

$C^{IV} = \{c^{IV}\}$, $c^{IV} = \langle o', o'', \text{type}\rangle$, $o' \in O$, $o'' \in O$, $o' \neq o''$ – hierarchical relationships (hierarchical structural constraints) "is a" defining class taxonomy (*type*=0), and "has part" / "part of" defining class hierarchy (*type*=1). The most abstract class is "*Thing*".

$C^V = \{c^V\}$, $c^V = (\{o\})$, $|\{o\}| \geq 2$, $o \in O$ – associative relationships ("one-level" structural constraints);

$C^{VI} = \{c^{VI}\}$, $c^{VI} = f(\{o\}, \{q\}) \rightarrow \text{True} \vee \text{False}$, $|\{o\}| \geq 0$, $|\{q\}| \geq 0$, $q \in Q$ – functional constraints referring to the names of classes and attributes.

$|c|$ – is a number of parameters included into a constraint (constraint cardinality).

Below, some example constraints are given:

- The attribute *costs* (q_1) belongs to the class *order* (o_1): $c^I_1 = (o_1, q_1)$;
- The attribute *costs* (q_1) belonging to the class *order* (o_1) may take positive values: $c^{II}_1 = (o_1, q_1, R^+)$;
- Instances of the class *standard operation* (o_2) can be compatible with instances of the class *resource* (o_3): $c^{III}_1 = (\{o_2, o_3\}, True)$;
- An instance of the class *order related operation* (o_4) can be a part of an instance of the class *order* (o_1): $c^{IV}_1 = \langle o_4, o_1, 1\rangle$;
- The order related operation (o_4) is an operation (o_5): $c^{IV}_1 = \langle o_4, o_5, 0\rangle$;
- An instance of the class *order related operation* (o_4) can be connected to an instance of the class *resource* (o_3): $c^V_1 = (o_2, o_3)$;
- The value of the attribute *cost* (q_1) of an instance of the class *order* (o_1) depends on the values of the attribute *cost* (q_1) of instances of the class *order related operation* (o_4) connected to that instance of the class *order* and on the number of such instances: $c^{VI}_1 = f(\{o_4\}, \{(o_4, q_1), (o_1, q_1)\})$.

A graphical interpretation of this model of the networked organization at macro level is presented in Fig. 3. The model contains one taxonomy (Thing, second level classes, third level classes) and two hierarchies (networked organization – networked organization member – Resource, and order – order related operation).

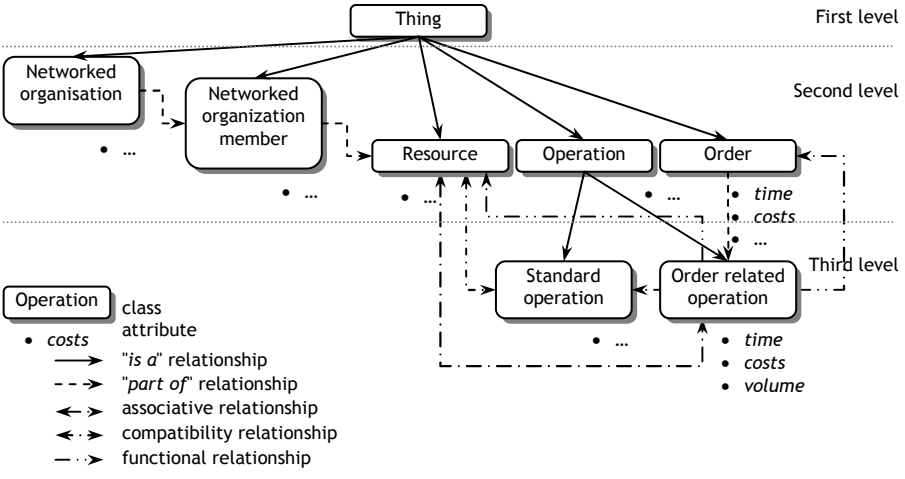


Fig. 3. Ontological model of a networked organisation at macro level

4 Implementation Standards and Protocols

One of the problems for this approach is assigning values to the attributes of objects acquired from different sources. In the presented work the access to the information sources is made in a unified way using Web-services as a de facto standard for enterprise application integration (EAI). Web-services translate information from information sources to the formalism of the system "KSNet". The services acquire information from information sources and transform it into the values of attributes of ontology.

According to the W3C consortium [11] a Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface that is described in a machine-processable format such as WSDL. Other systems interact with the Web service in a manner prescribed by its interface using messages, enclosed in a SOAP envelope. These messages are typically conveyed using HTTP, and normally comprise XML in conjunction with other Web-related standards.

The idea of Web-services arises from the concept of virtual organization. It can be said that "services are oriented to virtualization of resources" [12]. Services address discovery & invocation of persistent services. Internal implementation of services can be of any nature but their interfaces have to be standardized. In other words services must have:

- Standard interface definition mechanisms: multiple protocol bindings, multiple implementations, local/remote transparency for:
- Global naming & references,

- Registration & discovery,
- Authorization,
- Notification;
- Multiple hosting targets: J2EE, .NET, "C", etc. supporting:
- Lifetime management,
- Concurrency,
- Manageability.

The conceptual scenario of a Web-service is follows shown in Fig. 4. A service requester sends a request for an appropriate service ("Find Service") to the UDDI Registry. The Registry returns handle of the appropriate service to the requestor. Then the requestor sends a request to the service and receives the result from it.

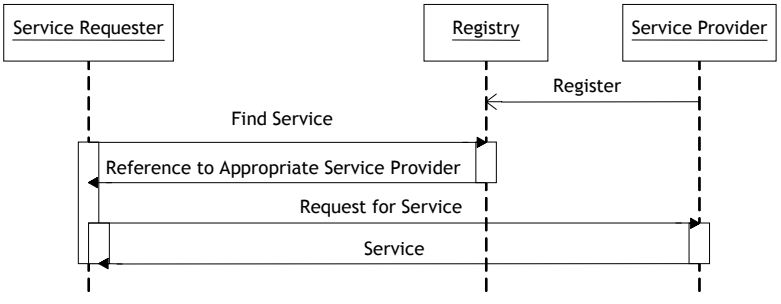


Fig. 4. Conceptual Web service operation scenario

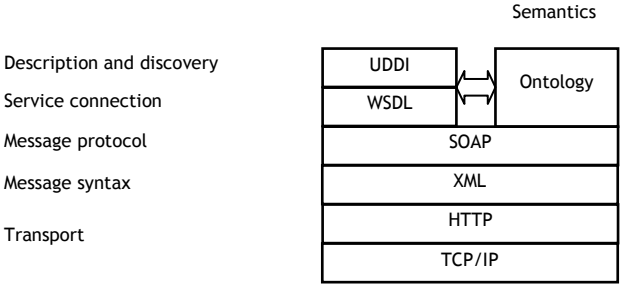


Fig. 5. Standards and protocols for open service framework

Current increasingly popular standards-based framework for accessing network applications is suggested by such companies as Microsoft [13], IBM [14], Sun [15] and others and supported by W3C consortium [11]. It includes (Fig. 5):

TCP/IP: Transmission Control Protocol/Internet Protocol – the suite of communications protocols used to connect hosts on the Internet;

HTTP: HyperText Transfer Protocol – the protocol used by the Internet;

XML: Extensible Markup Language – a specification developed by the W3C for a pared-down version of SGML, designed especially for Web documents;

SOAP: Simple Object Access Protocol – XML-based RPC protocol;

WSDL: Web Services Description Language – Interface Definition Language for open services;

UDDI: Universal Description, Discovery, & Integration – a "meta service" for locating open services by enabling robust queries against rich metadata [16].

In the presented approach the following kinds of information sources are distinguished:

- Sensors are physical devices that detect a signal, physical condition, etc; Web-services working with sensors have to support their outputs and interfaces;
- Databases are organized collections of information; Web-services that work with databases have to support SQL queries;
- Users may know large amount of information and can give it to the Web-service through graphical user interface.

The mechanisms of interaction between information sources and their assigned services are out of the scope of this research. Above regard the term "information source" will denote the information source itself and the Web-service assigned to it together.

The following two types of information sources are distinguished:

- Pull – information sources deliver information when the DSS requests it. A temperature sensor can be mentioned as an example of such information source. It measures temperature and DSS uses its Web-service interface when this information is needed.
- Push – information sources can send information to the Web-service of DSS. For example fire alarm sensor in case of fire can send this information to activate a corresponding scenario.

The scheme of retrieving information from the information sources is as follows. One or more attributes can request needed information from one Web-service that, in turn, can request this information from the one or more information sources (for example, a value of an attribute can be calculated based on information from several sources or sources can be alternative). When necessary the Web-service calculates the value(s) of the attribute(s) based on information from different information source(s).

5 Case Study

This section demonstrates application of the above approach in two areas: knowledge sharing within one company and knowledge sharing across a networked organisation.

5.1 Knowledge Sharing within One Company

In this case study it was necessary to access information about products and solutions stored in various sources (documents, databases, rule bases and Web sites) for an industrial company that has more than 300.000 customers in 176 countries supported by more than 50 companies worldwide with more than 250 branch offices and

authorised agencies in further 36 countries [17]. Among the major tasks that had to be solved the following should be outlined:

1) Keep existing facilities of the applications and avoid doubling of data;
2) Extend opportunities of fast provision of information about the company's products by new features (like free text search, feature prioritisation and other);
3) Provide multilingual interface;
4) Implement local and Web versions of the software;
5) Index existing documents against information stored in the databases.

To adopt the developed approach to the company's requirements the following tasks were solved:

1) Knowledge sources were selected and interfaces for accessing them were developed;
2) An ontological model, which is a part of the company's ontology, based on available structured data was built and extended by user-defined elements and synonyms;
3) Special methods to convert documents into machine readable formats were developed
4) An interface to other corporate databases was developed;
5) Documents were indexed against the ontological model vocabulary and the knowledge map was created;
6) Methods for calculation of the results relevance, fuzzy string comparison, and document ranking were developed.

The ontological model, which is a basis for corporate knowledge description, was built using structured information from databases and rule bases (Fig. 6). For this purpose a number of software modules were created that then were used for automatic creation of the ontology. It made possible to access all available information as if it was stored in a single knowledge base.

As a result a system was built that based on user (customer) requests activated appropriate knowledge sources (Web sites, documents, databases and rule bases) and provided access to them. The system was successfully tested within the company.

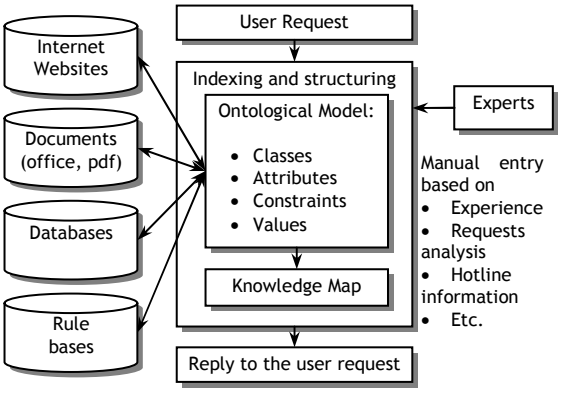


Fig. 6. Relation between application ontology and knowledge sources

5.2 Knowledge Sharing across a Networked Organisation

The FP6 project "Intelligent Logistics for Innovative Product Technologies" (ILIPT) is devoted to development of new methods and technologies to facilitate the implementation of a new manufacturing paradigm [18]. This new paradigm, "the 5-day car" will approach the building of 'cars to order' in a reduced time scale. ILIPT project will address the conceptual and practical aspects of delivering cars to customers only within several days after placing the order, the automotive industry's exciting and radical new business model [19]. One of the tasks of the ILIPT project is development of a common knowledge management platform to support interoperability within the "5-day car" production network. This will make it possible to accumulate, share, reuse and process knowledge across the "5-day car" production network that in turn can significantly help in increasing the supply chain effectiveness and in decreasing the lead time.

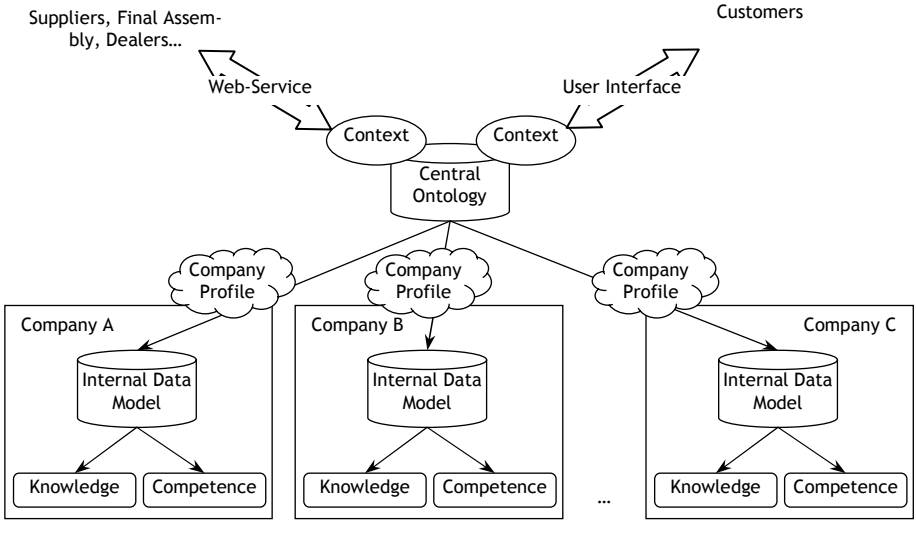


Fig. 7. Conceptual model of ontology-driven knowledge sharing

The conceptual model of the knowledge management platform is presented in Fig. 7. The ontology forms the core of the platform. It describes common entities (objects, facilities, products, processes, etc.) of the networked organisation members and relationships between them. In order for the ontology to be of reasonable size it includes only most generic common entities of the participating companies. For modern decision support systems, personalized support is important. Usually it is based on application of the profiling technology. The knowledge management platform provides an access to internal data models of the production network members through their competence profiles. As a result it is possible to treat all available knowledge and competencies as one distributed knowledge base. Each user (human or an information system) works on a particular problem or scenario represented via a context. Context is defined as any information that can be used to characterize the situation of an entity, where an entity is a person, place, or object that is considered relevant to the

interaction between a user and an application, including the user and applications themselves [20]. The context defines a narrow domain the user of the knowledge management platform works with. It may be characterised by a particular customer order, its time, requirements, etc.

This case study demonstrates creation of the ontological model for a networked organisation (a virtual production network). It was built using several source ontologies and setting relationships between their elements. Fig. 8 represents establishment of relationships between task & methods ontology and domain ontology "Configuration".

Summing up definitions of networked organisation configuration (e.g., [21, 22]) it can be defined as configuration of flows of products and services, finances and information between different stages from a supplier to a consumer / customer and managing operational activities of procurement and material releasing, transportation, manufacturing, warehousing and distribution, inventory control and management, demand and supply planning, order processing, production planning and scheduling, and customer service across the networked organisation.

The resulting networked organisation domain ontology is given in Fig. 9. The figure presents the class hierarchy for the classes of the taxonomy level following the root. Management concepts are constructed to cover various stages, functions, decisions, and flows.

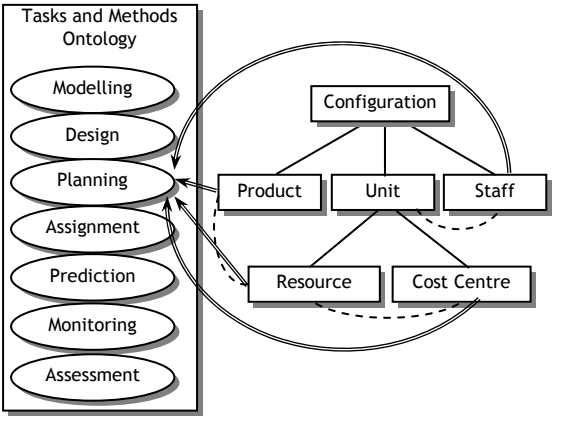


Fig. 8. Task & Method and Management ontologies (a fragment)

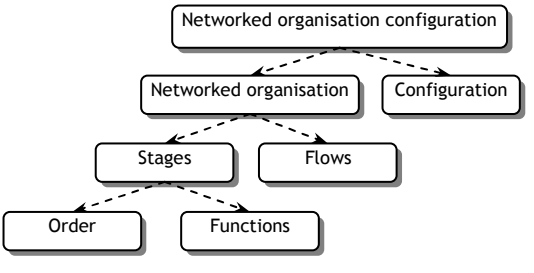


Fig. 9. Networked organisation configuration domain ontology: top-level classes view

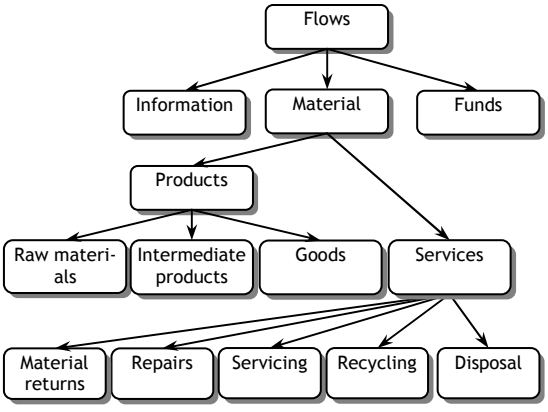


Fig. 10. Networked organisation flows: taxonomy

Networked organisation activities include *flow* of information, materials, and finances between different stages from suppliers to customer (Fig. 10). *Information flow* includes capacity, promotion plans, delivery schedules, sales, orders, inventory, quality; *material flow* contains row materials, intermediate products, finished goods, material returns, repairs, servicing, recycling, disposal; *finances flow* is made up of credits, consignment, payments [23]. Detailed specializations for products and services can be found in various product ontologies and classification systems (e.g., [24–26]) and mapped onto the presented classification level of the material flow.

Virtual production network as a networked organisation is a mechanism to integrate production functions taking place at the separate stages. Most of the *functions* (Fig. 11) happen within various stages, some of them cross the boundaries among several stages [27]. The functions operate on the networked organisation flows.

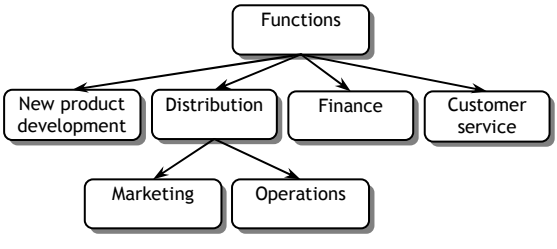


Fig. 11. Networked organisation functions: taxonomy

Part of the built ontological model presented in Fig. 12 focuses on the partner choice task. Since the problem considered is very complex a part of the ontology is given.

As a characteristic influencing networked organisation performance, cost is considered. In fact many cost items make up the total costs of the product required by the customer and the production, among them manufacturing costs, shipping costs, and

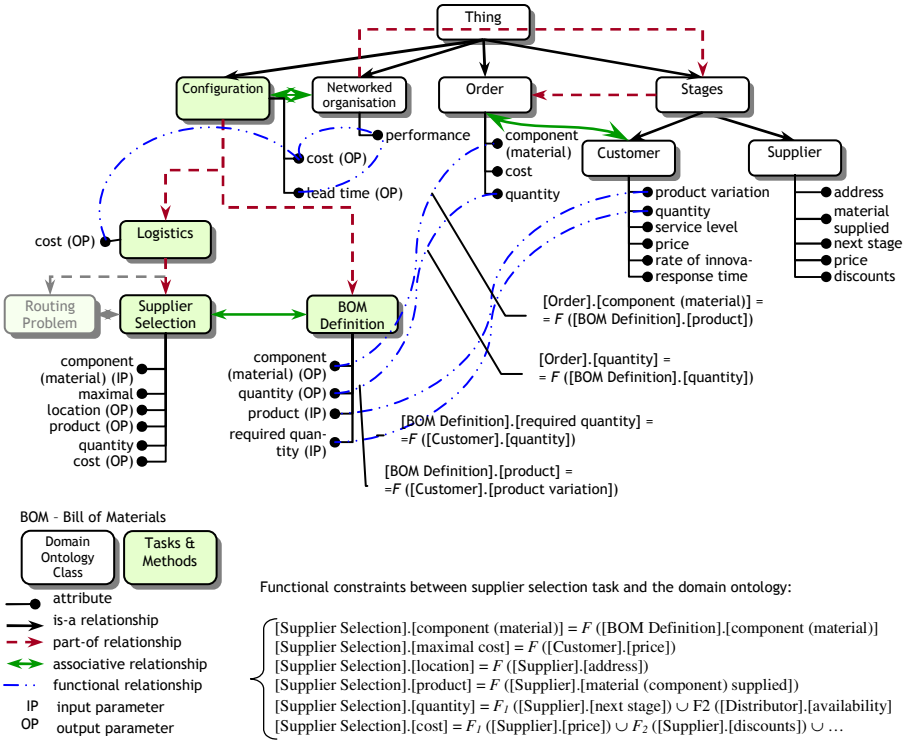


Fig. 12. Application ontology for networked organisation management (a fragment)

other are. This means that the complete ontological model includes all domain ontology classes that have an influence on costs. To simplify illustration interrelations between the domain ontology and the set of tasks are given by the example of the task of forming order for bill of materials (BOM). This task defines a set of materials and components that compose the product ordered by the customer.

The supplier selection task follows BOM definition task and has a set defined as input parameters. The task also takes into account maximal cost of the product that the customer is ready to pay, if any. Within the limits of the considered example the supplier selection task and the domain ontology are interrelated by the set of functional constraints shown in the bottom of Fig. 9.

Analogously, the supply chain performance depends on supply chain configuration cost combined with other influencing items:

$$[\text{Supply Chain}].[\text{performance}] = F_1([\text{Supply Chain Configuration}].[\text{cost}]) \cup F_2 \ \ldots$$

6 Conclusions

The paper presents an approach to semantic information integration for intelligent decision support in networked organisations. Usage of ontological knowledge description

made it possible to provide for common terminology and notation what, in turn, enabled integration of different tasks, constituting a common complex problem.

The approach has been tested in production related projects described in the section 4. One of them was implemented for an industrial company and was oriented to providing an access for users to different sources containing information and knowledge about company's products and services. The aim of other project was configuration of a BTO (build-to-order) production network consisting of several manufacturing facilities (suppliers).

Among the limitation of the approach the complexity of the common ontological model creation can be mentioned. However, the advantage of the ontological model is that it is a conceptual model of a high abstraction. Hence it can be defined for most general concepts and detailed concepts can be described only in the tasks.

Acknowledgements

The paper is due to the research carried out as a part of Integrated Project FP6-IST-NMP 507592-2 "Intelligent Logistics for Innovative Product Technologies" sponsored by European Commission, projects supported by the Russian Academy of Sciences # 16.2.35 of the research program "Mathematical Modelling and Intelligent Systems" and # 1.9 of the research program "Fundamental Basics of Information Technologies and Computer Systems", and projects funded by grants # 05 01 00151 and # 06 07 89242 of the Russian Foundation for Basic Research.

References

1. Laudon, K.C., Laudon, J.P.: Management Information Systems: Organisation and Technology in the Networked Enterprise (2000)
2. Lipnack, J., Stamps, J.: The Age of the Network: Operating Principles for the 21st century (1994)
3. Skyrme, D.J.: The Networked Organization (2003), http://www.skyrme.com/insights/1netorg.htm
4. Liu, M., Zhong, P., Meng, X., Liu, D., Cheng, H.: Knowledge-based integrated process management in lifecycle of product development. In: Sobolewsky, M., Cha, J. (eds.) Concurent Engineering: The Worldwide Engineering Grid, Proceedings of the 11th ISPE International Conference on Concurrent Engineering, pp. 181–186. Tsighua University Press, Springer (2004)
5. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., Haritatos, F.: Knowledge source network configuration approach to knowledge logistics. Int. J. of General Systems 32(3), 251–269 (2003)
6. Interoperability Research for Networked Enterprises Applications and Software, http://www.interop-noe.org
7. Semantic Web, http://www.semanticweb.org
8. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T.: Agent-based support of mass customization for corporate knowledge management. Engineering Applications of Artificial Intelligence 16(4), 349–364 (2003)
9. Baumgaertel, H.: Distributed constraint processing for production logistics. IEEE Intelligent Systems 15(1), 40–48 (2000)

10. Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W.: Enabling technology for knowledge sharing. AI Magazine 12(3), 36–56 (1991)
11. W3C consortium, http://www.w3.org 12
12. The Globus Project. An Open Grid Services Architecture, Globus Tutorial, Argonne National Laboratory (2002), http://www.globus.org/ogsa/deliverables/OGSA-January-2002-v3.pdf
13. Microsoft corporate, http://www.microsoft.com
14. IBM corporate, http://www.ibm.com
15. Sun corporate, http://www.sun.com
16. UDDI.org, http://www.uddi.org
17. Hinselmann, T., Smirnov, A., Pashkin, M., Chilov, N., Krizhanovsky, A.: Implementation of Customer Service Management System for Corporate Knowledge Utilization. In: Karagiannis, D., Reimer, U. (eds.) PAKM 2004. LNCS (LNAI), vol. 3336, pp. 475–486. Springer, Heidelberg (2004)
18. Stone, G., Miemczyk, J., Esser, R.: Making Build to Order a Reality: The 5-Day Car Initiative, Strengthening Competitiveness through Production Networks. A prospective from European ICT research projects in the field of "Enterprise Networking", pp. 26–37 (2005)
19. Intelligent Logistics for Innovative Product Technologies, http://www.ilipt.org
20. Dey, A.K., Salber, D., Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In: Moran, T.P., Dourish, P. (eds.) Context-Aware Computing, A Special Triple Issue of Human-Computer Interaction, vol. 16, Lawrence-Erlbaum, Mahwah (2001), http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf
21. Cooper, M.C., Lambert, D.M., Pagh, J.D.: Supply Chain Management: More Than a New Name for Logistics. The Int. J. of Logistics Management 8(1), 1–14 (1997)
22. Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E.: Designing and Managing the Supply Chain, Concepts, Strategies, and Case Studies. McGraw-Hill, New York (2000)
23. Chopra, S., Dougan, D., Taylor, G.: B2B e-Commerce Opportunities. Supply Chain Management Review, pp. 50–58 (May/June 2001)
24. UN Classifications Registry, United Nations Statistics Division (UNSD), http://unstats.un.org/unsd/cr/registry/default.asp
25. The UNSPSC Code (Universal Standard Products and Services Classification Code), DAML Ontology Library, Stanford University, http://www.ksl.stanford.edu/projects/¬DAML/UNSPSC.daml
26. The North American Industry Classification System (NAICS), http://www.census.gov/epcd/www/naics.html
27. Chopra, S., Meindl, P.: Supply Chain Management, Strategy, Planning, and Operation. Prentice Hall, Englewood Cliffs (2001)

HiLeX: A System for Semantic Information Extraction from Web Documents

Massimo Ruffolo[1,2] and Marco Manna[3]

[1] Exeura s.r.l.
[2] ICAR-CNR - Institute of High Performance Computing and Networking of the Italian National Research Council
[3] Department of Mathematics
University of Calabria, 87036 Arcavacata di Rende (CS), Italy
ruffolo@icar.cnr.it, manna@mat.unical.it
http://www.exeura.it

Abstract. Recognizing and extracting meaningful information from Web unstructured documents, taking into account their semantics, is an important problem of information and knowledge management. This paper describes HiLeX, a system implementing a novel logic-based approach to information extraction from unstructured documents. The approach adopted in the HiLeX system is founded on a new two-dimensional representation of documents, and heavily exploits DLP^+ - an extension of disjunctive logic programming for ontology representation and reasoning, which has been recently implemented on top of the *DLV* reasoning environment. Unlike previous systems, which are mainly syntactic, HiLeX combines both semantic and syntactic knowledge for a powerful information extraction. Ontologies, representing the semantics of information to be extracted, are encoded in DLP^+, while the extraction patterns are expressed using regular expressions and an ad hoc two-dimensional grammar. The execution of DLP^+ reasoning modules, encoding the grammar expressions, yields the actual extraction of information from the input document. HiLeX allows the semantic information extraction from both HTML pages and flat text documents by using synthetic and very expressive extraction patterns.

1 Introduction

HTML and flat text documents contain a huge amount of information arranged for human readers according to syntactic, semantic and presentation rules of a given language. That information tends to be practically unuseable, both for their vastness, and the lack of machine readability that makes existing information extraction systems unable to manage the actual knowledge that the information conveys.

Recognizing and extracting relevant information automatically from web documents, according to their semantics, is an important problem in the field of web information extraction.

In the recent literature a number of approaches for information extraction from unstructured documents have been proposed. An overview of the large body of existing literature and systems is given in [5,9,12,13,15]. The currently developed systems are

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 194–209, 2008.

mainly syntactic, and are not aware of the semantics of the information they are able to extract. They principally use pattern matching mechanism exploiting the underlying HTML syntactical structure and regular expressions on textual fragments contained between HTML tags.

In this work we present HiLeX, a logic-based system which combines both syntactic and semantic knowledge for a powerful information extraction from unstructured documents. Logic-based approaches for information extraction are not new [1,2], however, the approach we propose is original. Its novelty is due to:

- The two-dimensional representation of an unstructured document. A document is viewed as a Cartesian plane composed by a set of nested rectangular regions called *portions*. Each portion, univocally identified through the cartesian coordinates of two opposite vertices, contains a piece of the input document (*element*) annotated w.r.t. ontology.
- The exploitation of a logic-based knowledge representation language called DLP^+, extending *DLP* [10] with object-oriented features, including classes, (multiple) inheritance, complex objects, types. DLP^+ is well-suited for representation and powerful reasoning on ontologies; the language is supported by the DLV^+ system [4], implemented on top of *DLV* [6,7,8,14].
- The use of ontologies, encoded in DLP^+, describing the domain of the input document. A concept of the domain is represented by a DLP^+ class; each class instance is a *pattern* representing a possible way of writing the concept and is used to recognize and annotate an element contained in a portion.
- The employment of a new grammar, named HiLeX two-dimensional grammar, for specifying the (above mentioned) patterns. HiLeX grammar extends regular expressions for the representation of two-dimensional patterns (like tables, item lists, etc.), which often occur in web pages and textual tabular data. The patterns are specified through DLP^+ rules, whose execution yields the *semantic information extraction*, by associating (the part of the document embraced by) each portion to an element of the domain ontology.

It is worthwhile noting that, besides the domain ontologies, HiLeX system uses also a *core ontology*, containing (patterns for the recognition of) general linguistic elements (such as date, time, numbers, email, words, etc.); presentation elements (such as font colors, font styles, background colors, etc.); structural elements (such as table cell, item lists, paragraphs, etc.) which are not bound to a specific domain but occur generally.

The advantages of the HiLeX system over other existing approaches are mainly the following:

- The extraction of information according to their semantics and not only on the basis of their syntactic structure.
- The possibility to extract information in the same way from documents in different formats. The same extraction pattern can be used to extract data from both flat text and HTML documents. This feature is not due to a preliminary HTML-to-text translation, but it comes from the higher abstraction obtained from the transformation of the input document in a set of logical portions.

– The possibility to obtain a "semantic" classification of the input documents w.r.t. an ontology, which is much more accurate and meaningful than the syntactic classifications provided by existing systems (mainly based on counting the number of occurrences of some keywords). This feature opens the door to many relevant applications (e.g., emails classification and filtering, skills classification from curricula, extraction of relevant information from medical records, etc.).

Distinctive features of the HiLeX system, summarized above, allow a better digital contents management and fruition in different application field such as: e-health, e-entertainment, e-commerce, e-government, e-business.

The remainder of this work is organized as a by example explanation of the web information extraction approach implemented in the HiLeX system. In particular: section 2 shows the two-dimensional document representation idea, section 3 describes the DLP^{+} knowledge representation language, section 4 describes how ontologies are used to represent the semantics of information to be extracted, section 5 shows the *logic two-dimensional representation* of unstructured documents; section 6 describes the syntax and the semantics of the HiLeX two-dimensional pattern specification grammar, finally, section 7 describes the architecture of the HiLeX system focusing on the logic-based pattern matching method yielding the actual extraction of information from unstructured documents.

2 Two-Dimensional Format of Unstructured Documents

The two-dimensional representation of an unstructured document is the main notion, which the semantic information extraction approach implemented in the HiLeX system, is based on. This notion is founded on the idea that an unstructured document can be considered as a Cartesian plane composed by a set of nested rectangular regions called *portions*. Each portion, univocally identified through the Cartesian coordinates of two opposite vertices, contains a piece of the input document representing an *element* of the information to be extracted. Information elements, organized according to syntactic, presentation and semantic rules of a language recognizable by a human reader, can be *simple* or *complex*. *Simple* elements are, for example, characters, table cells, words (classified using their part-of-speech tag recognized using natural language processing techniques), dates, numbers; *complex* elements are, for example, phrases, item lists, tables, paragraphs, text boxes obtained as composition of other simple or complex elements.

To better explain the idea of portion consider the web page in Figure 1 (obtained from the Italian Yahoo financial portal) containing information about the stock exchange market. Suppose we would like to acquire, from this page, the table containing the stock index values and their variation (surrounded by a smooth etched box in Figure 1). A two-dimensional representation of data contained in the highlighted document region we are interested in (Figure 2), can be obtained by drawing on it an hypothetical Cartesian plane. Each element of the table can be identified, in that plane, by suitable rectangular regions (*portions*).

For instance, in Figure 2, the stock index name "Mib 30" is a simple element which is contained in the portion identified by $[(x_1, y_2),(x_2, y_3)]$. In the same way, the signed

Fig. 1. Financial Yahoo Page

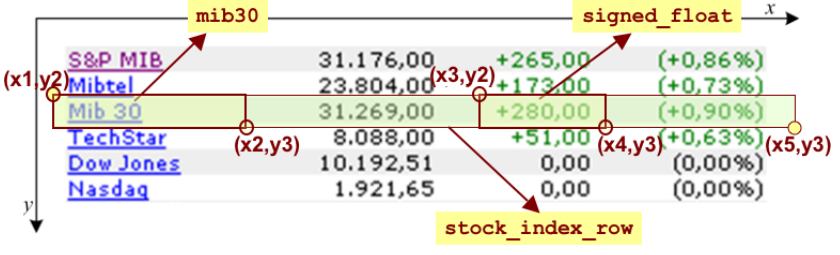


Fig. 2. Example of `portions`

float number representing the absolute variation of the "Mib 30" is contained in the portion $[(x_3, y_2),(x_4, y_3)]$. Since portions can be nested, the portion containing the complex element representing the concept of "stock index row" can be identified by the points $[(x_1, y_2),(x_5, y_3)]$ and so on.

3 The Knowledge Representation Language DLP^+

The DLP^+ [4] ontology representation language allows, in the HiLeX system, the representation of the semantics of information to be extracted the bottom-up reasoning capability on which pattern matching mechanisms is founded on.

DLP^+ is a powerful logic-based language which extends Disjunctive Logic Programming (DLP) [6] by object-oriented features. In particular, the language includes, besides the concept of relations, other notions coming from object-oriented world such as classes, objects (class instances), object-identity, complex-objects, (multiple) inheritance, and the concept of modular programming by means of *reasoning modules*. This makes DLP^+ a complete ontology representation language supporting sophisticated reasoning capabilities.

The DLP^+ language is implemented in the DLV^+ system, a cross-platform development environment for knowledge modeling and advanced knowledge-based reasoning. The DLV^+ system [4] permits to easily develop real world complex applications and allows advanced reasoning tasks in a user friendly visual environment. DLV^+ seamlessly integrates the DLV [6] system exploiting the power of a stable and efficient Answer Set Programming solver (for further background on DLV and DLP^+ see [4,6]).

In the HiLeX system the DLP^+ language is heavily exploited for the formal representation of the semantics of information to be extracted (employing suitable ontologies). Furthermore, DLP^+ allows the encoding of the *logic two-dimensional representation* of unstructured documents. Finally, DLP^+ reasoning modules (which are specialized DLP^+ logic programs) allow the exploitation of the the bottom-up reasoning capability, and thus the implementation of the logic-based pattern matching method yielding the actual semantic information extraction.

4 Ontology Representation in HiLeX

In the HiLeX system the elements of information to be extracted are modeled by using the DLP^+ class *element* which is defined as follows:

```
class element (type: expression_type,
   expression: string, label: string).
```

The three attributes have the following meaning:

- `expression`: holds a string representing the pattern specified by regular expressions or by the HiLeX two-dimensional grammar (described in detail in the section 6), according to the `type` property. Patterns contained in these attributes are used to recognize the elements in a document.
- `type`: defines the type of the expression (i.e. `regexp_type`, `hilex_type`).
- `label`: contains a description of the element in natural language.

The element class is the common root of two kind of ontologies, the `core ontology` and the `domain ontologies`. Every pattern encoding information to be extracted is represented by an instance of a class belonging to these ontologies.

In the following the structure of core and domain ontologies are described in greater detail.

4.1 The Core Ontology

The core ontology is composed of three parts. The first part represents general simple elements describing a language (like, e.g., alphabet symbols, lemmas, Part-of-Speech, regular forms such as date, e-mail, etc.). The second part represents elements describing presentation styles (like, e.g., font types, font styles, font colors, background colors, etc.). The third part represents structural elements describing tabular and textual structures (e.g. table cells, table columns, table rows, paragraphs, item lists, texture images, text lines, etc.). The core ontology is organized in the class hierarchy shown below:

```
class linguistic_element isa {element}.
   class character isa {linguistic_element}.
      class number_character isa {character}.
      ...
   class regular_form isa {linguistic_element}.
      class float_number isa {regular_form}.
      ...
   class italian_lexical_element isa {linguistic_element}.
   class english_lexical_element isa {linguistic_element}.
       class english_lemma isa {english_lexical_element}.
      ...
   class spanish_lexical_element isa {linguistic_element}.
   ...
class presentation_element isa {element}.
   class font_type isa {presentation_element}.
   ...
class structural_element isa {element}.
   class table_cell isa {structural_element}.
   class separator isa {structural_element}.
   ...
```

Examples of instances of the `float_number` class are:

```
unsigned_float_number: float_number (type: regexp_type,
    expression:"(\d{1,3}(?>.\d{3})*,\d+)",
    label: "RegExp for unsigned float number").

signed_float_number: float_number (type: regexp_type,
    expression:"([+-]\s*\d{1,3}(?>.\d{3})*,\d+)",
    label: "RegExp for signed float number").

percentage: float_number (type: regexp_type,
    expression:"(\(?(?>[+-])?(?>(?>100(?>,0+)?)|
                (?:\d{1,2}(?>,\d+)?))%\)?)",
    label: "RegExp for percentage").
```

When in a document the regular expression characterizing a particular kind of float number is recognized, a document portion is generated and annotated w.r.t. the corresponding class instance.

4.2 Domain Ontologies

Domain ontologies contain simple and complex elements of a specific knowledge domain. The distinction between core and domain ontologies allows to describe knowledge in a modular way. When a user needs to extract data from a document regarding a specific domain, he can use only the corresponding domain ontology. The modularization improves the extraction process in terms of precision and overall performances. Referring to the example of the stock index market domain, elements representing related concepts can be organized as follows:

```
class stock_market_domain isa {element}.
   class stock_index isa {stock_market_domain, linguistic_element}.
   class stock_index_cell isa {stock_market_domain, structural_element}
   class stock_index_row isa {stock_market_domain, structural_element}.
   class stock_index_table isa {stock_market_domain, structural_element}.
   class index_value isa {stock_market_domain, regular_form}.
```

Examples of instances of the `stock_index` class are:

```
mibtel: stock_index (type: regexp_type,
   expression: ''Mibtel'').
mib30: stock_index (type: regexp_type,
   expression: ''Mib30'').
dowJones: stock_index (type: regexp_type,
   expression: ''Dow Jones'').
```

When a regular expression characterizing a stock index is recognized in a document, a portion is generated and annotated w.r.t. the corresponding class instance.

5 Logic Two-Dimensional Representation of Unstructured Documents

As pointed out in section 2, elements are located inside rectangular regions of the input document called `portions`. Document portions, and the enclosed elements, are represented in DLP^{+} by using the class *point* and the relation *portion*

```
class point (x: integer, y: integer).
relation portion (p: point, q: point, elem: element).
```

Each instance of the relation `portion` represents the relative rectangular document region. It relates the two points identifying the region, expressed as instances of the class `point`, and an ontology element, expressed as instance of the class `element`. The set of instances of the `portion` relation constitute the *logic two-dimensional representation* of an unstructured document. This DLP^{+} encoding allows to exploit the two-dimensional document representation on which the semantic information extraction approach proposed in this paper is based on.

6 A Two-Dimesional Grammar for Extraction Patterns Specification

The internal representation of extraction patterns, in the HiLeX system, is obtained by means of a two-dimensional grammar, founded on picture languages [3,11], and allowing the definition of very expressive target patterns. Each pattern represents a two-dimensional composition of portions annotated w.r.t. the elements defined in the ontology. The syntax of the HiLeX two-dimensional grammar is presented in the following.

```
NEW_ELEMENT → GENERALIZATION | RECURRENCE | CHAIN | TABLE
GENERALIZATION → GEN1 | GEN2 | GEN3
GEN1 → generalizationOf (arg: ARG1)
GEN2 → orContain_generalizationOf (arg: ARG1, inArg: ARG1, condition: CND)
GEN3 → andContain_generalizationOf (arg: ARG1, inArg: ARG1, condition: CND)
CND → coincident | notCoincident | null
RECURRENCE → recurrenceOf (arg: ARG3, range: RANGE, dir: DIR)
CHAIN → CHAIN1 (arg: ARG2, dir: DIR, sep: SEP)
CHAIN1 → sequenceOf | permutationOf
TABLE → TAB1 (arg: ARG2, range: RANGE, dir: DIR, sep: SEP)
TAB1 → sequenceTableOf | permutationTableOf
ARG1 → ARG2 | ARG3
ARG2 → [ LIST ]
ARG3 → BASE_ELEM
LIST → ARG3 , ARG3 LIST1
LIST1 → , ARG3 LIST1 | ε
RANGE → < NUM , NUM > | NUM | + | *
DIR → vertical | horizontal | both
SEP → ARG3 | null
```

According to the HiLeX grammar, a portion annotated w.r.t. a NEW_ELEMENT can be obtained by applying the composition language constructs to portions annotated w.r.t. basic ontology elements (BASE_ELEM i.e. instances of ontology classes). The semantics of each construct, together with some examples of usage, are presented in the following.

GENERALIZATION: A portion annotated to basic ontology element (BASE_ELEM) can be re-annotated to the new ontology element (NEW_ELEMENT), by using the generalizationOf operator. The effect of this operator is a semantic rewriting generalizing the portion annotation.

Example 1. Consider the HTML document presented in section 2 whose elements are properly modelled in the core and domain ontologies. Let unsigned_float_number be an instance of the float_number class defined in the core ontology. A portion annotated as unsigned_float_number is re-annotated w.r.t. absolute_index_value by using the following expression:

```
absolute_index_value: index_value (type:hilex_type,
    expression:"generalizationOf (arg: unsigned_float_number)",
    label:"Absolute value of a stock index" ).
```

The HiLeX grammar constructs orContain_generalizationOf and andContain_generalizationOf allow to define new annotations of existing portion on the basis of the semantics of contained portions. In particular, using the orContain_generalizationOf, a portion already annotated w.r.t. the arg elements, is re-annotated w.r.t. the class instance, in which the rule appears, if the portion itself contains at least one portion annotated w.r.t. one of the inArg arguments.

RECURRENCE: A portion annotated w.r.t. a NEW_ELEMENT, obtained by means of the recurrenceOf operator, consists in the *null portion* (i.e. a portion without annotation

having overlapped vertex along a coordinate) or in the concatenation, along a given direction, of a fixed number of portions annotated w.r.t. the same BASE_ELEM.

Example 2. Using the HiLeX recurrenceOf construct, a separator between two elements, contained in a document, can be defined as an instance of the separator class, constituted by a *null portion* or by the concatenation, in the horizontal direction, of an undefined number of portions annotated w.r.t. the blank_char element (Figure 3), defined as an instance of the core ontology character class as shown in the following.

```
sep_01: separator (type: hilex_type, expression:
   "recurrenceOf (arg: blank_char, range: *, dir: horizontal)",
      label: "Blank characters separator").
```

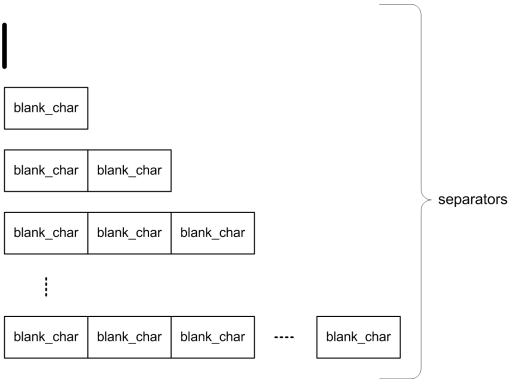


Fig. 3. Example of recurrence

CHAIN: A portion annotated w.r.t. a NEW_ELEMENT by using the sequenceOf and permutationOf operators, constitutes a chain of portions annotated w.r.t. BASE_ELEMs. In particular, a portion obtained by the application of the sequenceOf operator is a concatenation of at least two portions annotated w.r.t. BASE_ELEMs in a given direction and a fixed order, whereas a portion obtained by using the permutationOf operator is a concatenation of at least two portions annotated w.r.t. BASE_ELEMs in a given direction, without an established order.

Example 3. A table row containing stock index variations can be represented using the HiLeX construct sequenceOf in the following:

```
stock_index_row_01: stock_index_row( type:hilex_type,
    expression:"sequenceOf( arg: [stock_index,
        absolute_index_value, absolute_index_variation,
        percentage_index_variation],
        dir:horizontal, sep:sep_01 )",
    label:"Row containing stock index variations" ).
```

The Figure 4 shows the portion annotated w.r.t an instance of the stock_index_row class. It is constituted by an ordered sequence, in the horizontal direction, of portions annotated w.r.t. any of the instances of the stock_index class, and the unsigned_float,

signed_float and percentage instances. Between each couple of portions could be present a portion annotated w.r.t the element sep_01, an instance of the separator class, defined in the example 2. This expression considers only the semantics of the portions and their spatial positioning. No reference to the underlying document structure is required to recognize the concept of stock_index_row.

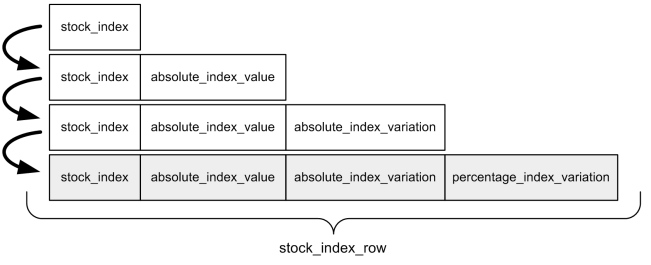


Fig. 4. Example of chain

TABLE: A portion annotated w.r.t. a NEW_ELEMENT can be defined by using the sequenceTableOf or permutationTableOf HtLεX operators, as a table of portions annotated w.r.t. BASE_ELEMs.

A portion, obtained from the sequenceTableOf operator, is composed by portions having a fixed composition along a direction, repeated a certain number of times along the other direction, whereas, a portion obtained from the permutationTableOf operator is composed by portions having an unordered composition along a direction, repeated with the same structure a fixed number of times along the orthogonal direction. This construct allows to recognize table in both HTML and text documents. In fact, portions provide an abstract representation of unstructured documents independent from the document format.

Example 4. The figure 5 depicts a portion annotated w.r.t. an instance of the stock_index_table class obtained by using the sequenceTableOf HtLεX grammar construct as shown in the following:

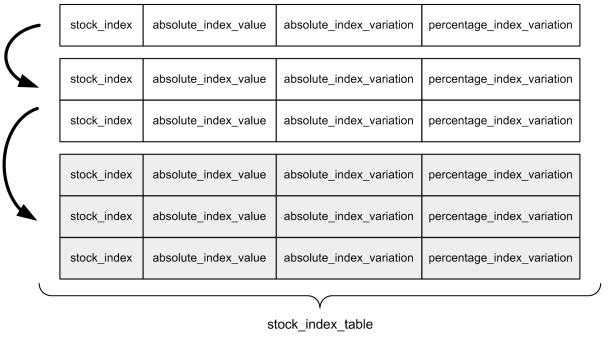


Fig. 5. Example of table

```
stock_index_table_01:stock_index_table( type: hilex_type,
    expression:"sequenceTableOf( arg: [stock_index,
        adsolute_index_value, absolute_index_variation,
        percentage_index_variation],
        range:<2,5>, dir:vertical, sep:sep_01 )",
    label:"table containing stock_index_row" ).
```

The instance `stock_index_table_01` represents a table of stock index variations composed by a vertical sequence of at least 2 and at most 5 rows. Each row is a sequence of other portions annotated w.r.t. instances of the class `stock_index`, and the `unsigned_float`, `signed_float` and a `percentage` (i.e. a `stock_index_row`) instances.

7 The HiLeX System

The architecture of the HiLeX system is depicted in figure 6. The semantic information extraction approach can be viewed as a process composed of four main steps: knowledge representation, document preprocessing, pattern matching and pattern extraction. Each step is executed by means of the architectural modules described in the next subsections.

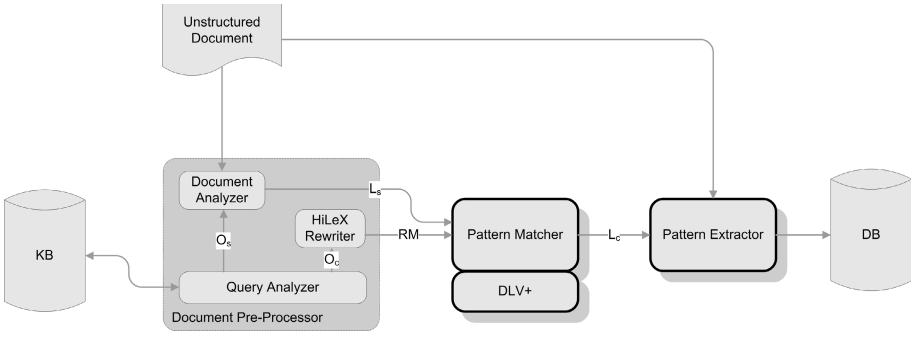


Fig. 6. The architecture of the HiLeX System

7.1 Knowledge Base

The Knowledge Base (KB) stores, using the *DLV*$^+$ system persistency layer, core and domain ontologies, arranged as described in the section 4, and the logic two-dimensional representation of unstructured documents (section 5). The KB provide an API containing methods aimed at handling ontology querying and at assisting the pattern matching.

7.2 Document Preprocessor

The document preprocessor takes as input an unstructured document, and a query containing the class instances names representing the information that the user wishes to extract. After the execution the document preprocessor returns the logic document

representation and a set of reasoning modules constituting the input for the pattern recognizer. The document preprocessing is performed by the three sub-modules described in the following.

Query Analyzer. This module takes as input the user query and explores the ontologies in order to identify patterns for the extraction process. The output of the query analyzer are two sets of couples (*class instance name*, *pattern*). The first set (O_s) contains couples in which instances are characterized by patterns represented by regular expressions (simple elements), whereas in the second set (O_c) patterns are expressed using the HiLeX pattern representation grammar (complex elements). The set O_s is the input for the document analyzer and the set O_c is the input for the rewriter module.

Document Analyzer. The input of this module is an unstructured document and the set of couples O_s. The document analyzer is able to recognize regular expressions, applying pattern matching mechanisms, to detect simple elements constituting the document and for each of them generates the relative *portion*. At the end of the analysis this module provides the *logic document representation* L_s which is a uniform abstract view of different document formats.

HiLeX Rewriter. The input for this module is the set of couples O_c containing the extraction patterns expressed by means of the HiLeX two-dimensional grammar. Each pattern is translated in a set of logical rules implemented in a DLP^+ reasoning modules (RM) which are to be executed by the DLV^+ system. The translation allows the actual semantic information extraction from unstructured documents performed by the pattern matcher module.

More in detail, the translation is based on predefined operators able to manipulate portions used to encode the HiLeX grammar expressions in DLP^+. Each pattern is rewritten in a DLP^+ reasoning module as a set of rules employing the following basic operators able to manipulate points and portions.

```
relation strictFollow(p1: point, q1: point,
         elem1: element, p2: point, q2: point, elem2: element).
relation strictBelow(p1: point, q1: point,
         elem1: element, p2: point, q2: point, elem2: element).
relation minContain (p1: point, q1: point,
         elem1: element, p2: point, q2: point, elem2: element).
relation horizontalRecurrence(p: point,
         q: point, elem: element, min: integer, max: integer).
relation verticalRecurrence(p: point,
         q: point, elem: element, min: integer, max: integer).
```

The `strictFollow` operator, for example, is implemented by means of the DLP^+ rule presented in following:

```
strictFollow (P1, Q1, E1, P2, Q2, E2) :-
   portion (p: P1, q: Q1, elem: E1),
   portion (p: P2, q: Q2, elem: E2),
   P1: point (y: YP), Q1: point (x: X, y: YQ),
   P2: point (x: X, y: YP), Q2: point (y: YQ).
```

The semantics of the five basic operators is intuitively given in Figure 7.

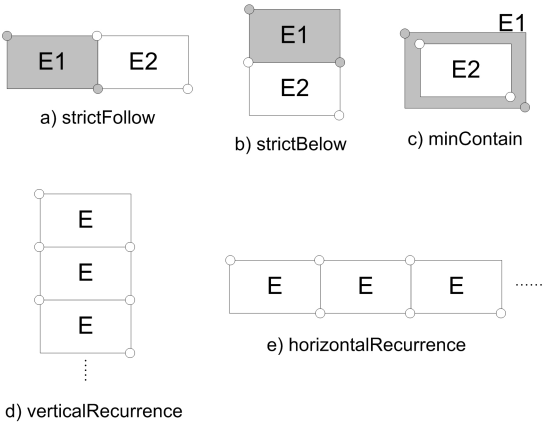


Fig. 7. Basic operators

The table containing the stock index variations, incorporated in the page presented in section 2, can be extracted using the pattern presented in the example 4. The corresponding DLP^{+} rewriting is shown below.

```
module(stock_index_table_01){
    portion(p:P1, q:Q7, elem:row_of_stock_index_table_01):-
        strictFollow(p1:P1, q1:Q1, elem1:E1,
                     p2:P2, q2:Q2, elem2:sep_01),
        strictFollow(p1:P2, q1:Q2, elem1:sep_01,
                     p2:P3, q2:Q3, elem2:absolute_index_value),
        strictFollow(p1:P3, q1:Q3, elem1:absolute_index_value,
                     p2:P4, q2:Q4, elem2:sep_01),
        strictFollow(p1:P4, q1:Q4, elem1:sep_01,
                     p2:P5, q2:Q5, elem2:absolute_index_variation),
        strictFollow(p1:P5, q1:Q5, elem1:absolute_index_variation,
                     p2:P6, q2:Q6, elem2:sep_01),
        strictFollow(p1:P6, q1:Q6, elem1:sep_01,
                     p2:P7, q2:Q7, elem2:percentage_index_variation),
        instanceOf(E1,stock_index).
    portion(p:P, q:Q, elem:stock_index_table_01):-
        verticalRecurrence(p:P, q:Q, elem:stock_index_table_row_01,
            min:2, max:5).}
```

The new portion having the semantic of table of stock index variation, whose structure satisfies the extraction pattern, is recognized by applying rules contained in the reasoning module shown above. These rules exploit the *logic two-dimensional representation* of unstructured document.

7.3 Pattern Matcher

The pattern matcher is founded on the DLV^{+} system. It takes as input the logic two-dimensional document representation (L_s) and the set of reasoning modules (RM)

containing the translation of the HiLeX patterns in term of logic rules and recognize new complex elements. The output of this step is the *augmented logic two-dimensional representation* (L_c) of an unstructured document in which new document regions, containing more complex elements (e.g. table having a certain structure and containing certain concepts, phrases having a particular meaning, etc.), are identified.

The logic-based pattern matching mechanism implemented in this module exploits the translation of extraction patterns performed by the HiLeX rewriter module.

7.4 Pattern Extractor

This module takes in input the augmented logic representation of a document (L_c) and allows the acquisition of element instances (semantic wrapping) and/or the document classification w.r.t. the ontologies classes.

The result of the extraction process, on the page presented in section 2, is graphically shown in Figure 8. Figure 8 (a) depicts portions identified using patterns represented by regular expressions. Figure 8 (b) and (c) show portions identified by the pattern matcher exploiting the logic representation of the HiLeX grammar expressions.

(a)

S&P MIB	31.176,00	+265,00	(+0,86%)
Mibtel	23.804,00	+173,00	(+0,73%)
Mib 30	31.269,00	+280,00	(+0,90%)
TechStar	8.088,00	+51,00	(+0,63%)
Dow Jones	10.192,51	0,00	(0,00%)
Nasdaq	1.921,65	0,00	(0,00%)

(b)

S&P MIB	31.176,00	+265,00	(+0,86%)
Mibtel	23.804,00	+173,00	(+0,73%)
Mib 30	31.269,00	+280,00	(+0,90%)
TechStar	8.088,00	+51,00	(+0,63%)
Dow Jones	10.192,51	0,00	(0,00%)
Nasdaq	1.921,65	0,00	(0,00%)

(c)

S&P MIB	31.176,00	+265,00	(+0,86%)
Mibtel	23.804,00	+173,00	(+0,73%)
Mib 30	31.269,00	+280,00	(+0,90%)
TechStar	8.088,00	+51,00	(+0,63%)
Dow Jones	10.192,51	0,00	(0,00%)
Nasdaq	1.921,65	0,00	(0,00%)

Fig. 8. Portions Extracted from the Yahoo Page

It is noteworthy that patterns are very synthetic and expressive. Moreover, patterns are general in the sense that they are independent from the document format. This last peculiarity implies that the extraction patterns, presented above, are more robust w.r.t. variations of the page structure than extraction patterns defined in the previous approaches. For example, the table containing the stock index variations could appear wherever in the page. Furthermore, the same extraction patterns can also be used to extract information from flat text having the structure depicted in figure 9. The result of the extraction process on flat text is depicted in Figure 9 (a), (b), (c) having the same structure of Figure 8.

Acquired instances can be stored in a DLP^+ ontology, a relational database, an XML database. So, extracted information can be used in other applications and more powerful query and reasoning task are possible on them. The extraction process causes the annotation of the documents w.r.t. the ontologies concepts. This feature can enable, for example in document management contexts the semantic classification.

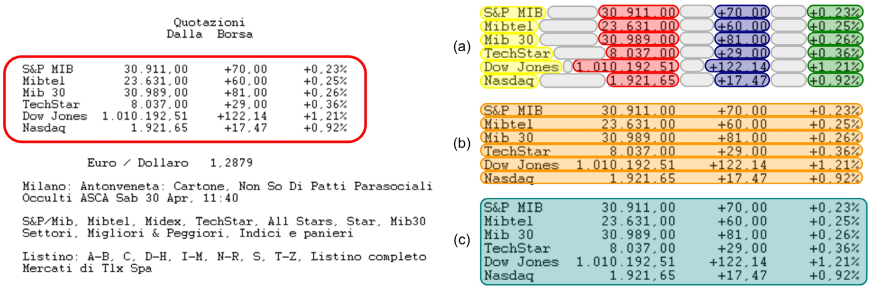


Fig. 9. Flat Text Version of the Yahoo Page

8 Conclusions and Future Works

This work presents a novel, concrete, powerful and expressive approach to information extraction from web documents. The approach, implemented in the HtLeX system, is grounded on two main ideas:

- The representation, by means of the DLP^{+} ontology representation language having solid theoretical foundations, of the semantics of information to extract.
- The logic two-dimensional representation of documents allowing the definition of extraction patterns expressed by the HtLeX two-dimensional grammar.

Thanks to these ideas, the approach constitutes an enhancement in the field of Web information extraction. Unlike previous approaches, the same extraction patterns can be used to extract information, according to their semantics, from both HTML and flat text documents. Furthermore, the HtLeX system can be used to implement a new generation of semantic wrappers. Many functions that will be available in the future "semantic web"technologies are turning into reality today with the HtLeX system.

Currently, consolidation of the approach is ongoing and its theoretical foundations are under investigation and improvement. Future work will be focused on the consolidation and extension of the HtLeX two-dimensional grammar, the investigation of the computational complexity issues from a theoretical point of view, the extension of the approach to PDF as well as other document formats, the exploitation of natural language processing techniques aimed at improve information extraction from flat text documents.

References

1. Baumgartner, R., Flesca, S., Gottlob, G.: Declarative information extraction, web crawling, and recursive wrapping with lixto. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 21–41. Springer, Heidelberg (2001)
2. Baumgartner, R., Flesca, S., Gottlob, G.: Visual web information extraction with lixto. The VLDB Journal, 119–128 (2001)

3. Chang, S.-K.: The analysis of two-dimensional patterns using picture processing grammars. In: STOC 1970. Proceedings of the second annual ACM symposium on Theory of computing, pp. 206–216. ACM Press, New York (1970)
4. Dell'Armi, T., Leone, N., Ricca, F.: Il linguaggio dlp+. Internal report, Exeura s.r.l (June 2004)
5. Eikvil, L.: Information extraction from world wide web - a survey. Technical Report 945, Norweigan Computing Center (1999)
6. Eiter, T., Faber, W., Leone, N., Pfeifer, G.: Declarative Problem-Solving Using the DLV System. In: Minker, J. (ed.) Logic-Based Artificial Intelligence, pp. 79–103. Kluwer Academic Publishers, Dordrecht (2000)
7. Eiter, T., Leone, N., Mateis, C., Pfeifer, G., Scarcello, F.: A deductive system for nonmonotonic reasoning. In: Logic Programming and Non-monotonic Reasoning, pp. 364–375 (1997)
8. Faber, W., Pfeifer, G.: Dlv homepage (1996)
9. Feldman, R., Aumann, Y., Finkelstein-Landau, M., Hurvitz, E., Regev, Y., Yaroshevich, A.: A comparative study of information extraction strategies. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 349–359. Springer, Heidelberg (2002)
10. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9(3/4), 365–386 (1991)
11. Giammarresi, D., Restivo, A.: Two-dimensional languages. In: Salomaa, A., Rozenberg, G. (eds.) Handbook of Formal Languages, Beyond Words, vol. 3, pp. 215–267. Springer, Berlin (1997)
12. Kuhlins, S., Tredwell, R.: Toolkits for generating wrappers – a survey of software toolkits for automated data extraction from web sites. In: Aksit, M., Mezini, M., Unland, R. (eds.) NODe 2002. LNCS, vol. 2591, pp. 184–198. Springer, Heidelberg (2003)
13. Laender, A., Ribeiro-Neto, B., Silva, A., Teixeira, J.: A brief survey of web data extraction tools. In: SIGMOD Record, vol. 31 (June 2002)
14. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning (2004)
15. Rosenfeld, B., Feldman, R., Fresko, M., Schler, J., Aumann, Y.: Teg: a hybrid approach to information extraction. In: Grossman, D., Gravano, L., Zhai, C., Herzog, O., Evans, D.A. (eds.) CIKM, pp. 589–596. ACM, New York (2004)

Skill-Profile Matching with Similarity Measures

Ernst Biesalski and Andreas Abecker

Research Center for Information Technologies,
Haid-und-Neu-Straße 10-14, 76131 Karlsruhe
Ernst.biesalski@googlemail.com, abecker@fzi.de

Abstract. At DaimlerChrysler's truck plant in Wörth / Rhein, we are currently implementing a comprehensive IT solution for integrated and synergistic processes in personnel development. In this paper, we sketch some ontology-based software modules – as well as their interdependencies and synergies – which support streamlined and integrated, comprehensive personnel-development processes. A central element in the software architecture is ontology-based similarity assessment for skill-profile matching which is exemplarily discussed for software-supported project staffing.

Keywords: Knowledge Management, Ontology, Skill Management.

1 Motivation

During the next ten years, Human Resource Management (HRM) departments in large companies in Germany and most parts of Europe will face radically challenges and tasks. Already today, demographic studies and prognoses show clearly, that in the long-term, the number of young people will significantly decrease [1]. Even with a realistically estimated amount of immigration, the share of citizen under 20 years will reduce from 20% (2001) to 16% (2050) of the overall population while the share of people above 60 years increases from about 25% to about 33%. Consequently, the working population will run through a continuous aging process (in the average), and from approx 2015 on, the number of persons available for employment will more and more run short. Under such conditions, a coordinated, long-term personnel development strategy gains increasing importance. Such a strategic personnel development must be part of a comprehensive HRM strategy which should in turn be embedded into an overall, holistic Knowledge Management (KM) approach (cp. [2]).

From the IT point of view, HRM departments mainly use IT applications for the management of personnel data (standing data, performance reviews), for junior employee development, or for training planning, seldom also for assessment of training needs. Real-world system landscapes are often characterized by manifold heterogeneous systems, evolved over time, showing pretty non-uniform features – which hinders interoperability of those applications. Further problems come from massively redundant data storage, as well as complex interfaces between systems. Since such system landscapes are typically a combination of standard software and proprietary developments of the HR software department, they seldom support an integrated personnel

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 210–218, 2008.

development approach, i.e., a coordinated behavior of different applications. Modern views on knowledge and skills of employees are normally not realized. The purpose of such systems is to manage the single employee, not to model and manage an integrated view on employee, tasks and organizational context.

At DaimlerChrysler's truck plant in Wörth / Rhine, we are currently developing such an *integrated* system and process landscape. In Section 2, we sketch the respective software architecture. In Section 3, we focus on one module of the system, designed for supporting project staffing. At the hand of this example, we discuss in more detail the ontology-based matching of skill profiles – which is a central functionality also for the other modules. Finally, in Section 4, we briefly summarize, sketch some related work and report on the current implementation status of the system.

2 An Application Framework for Personnel Development

We propose an integrated software and process framework for personnel development (PD Framework) as depicted in Figure 1. It contains business processes for personnel development [3], a Human Resource Data Warehouse (HR DWh) which integrates data from different legacy systems, an ontology-based employee-skill database, as well as different new application modules.

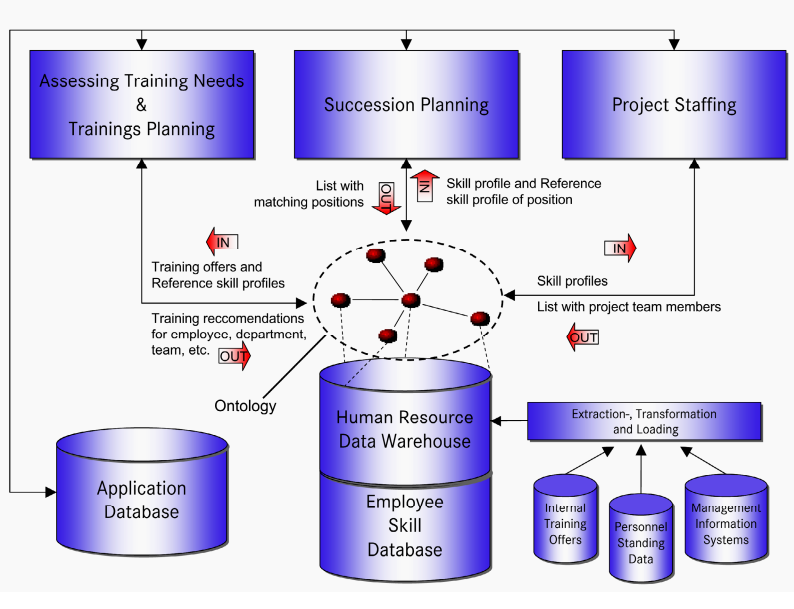


Fig. 1. Overall Perspective on Personnel Development Software Framework

In this article, we focus on the application module for project staffing. A core idea of this and all the other modules is the ontology-based modeling of employees' skill profiles. This is based upon an ontology which formalizes the former skill catalog that

describes all different personal skills occurring or required in the company (cp. Figure 2). Each software module employs an ontology-based matching procedure which is able to compare skill profiles (i.e. bundles of skills which characterize an employees' knowledge, skills, and qualifications, or, the competences required for a specific job, respectively). For the **"Succession Planning"** module, this means to compare the to-be skill profile of an open position with the as-is profiles of a number of employees – in order to find out the most suited candidate. To this end, we compare bundles of ontology instances, as indicated in Figure 3.

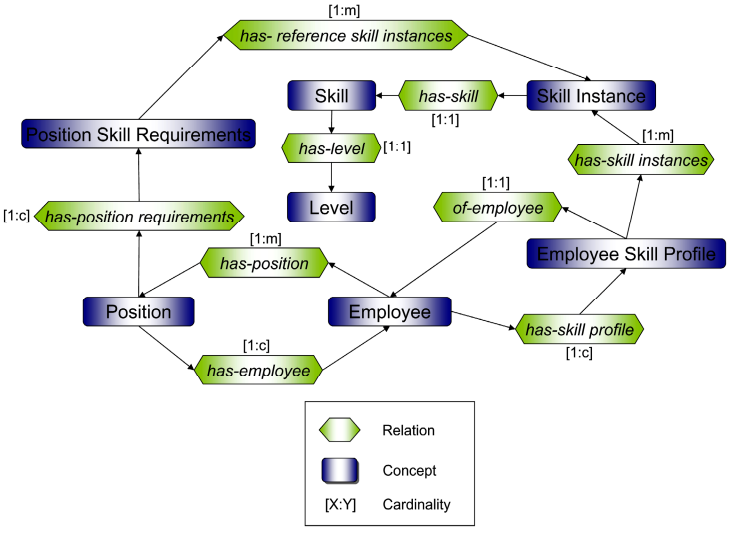


Fig. 2. Top-Level Structure of Skill Ontology (simplified)

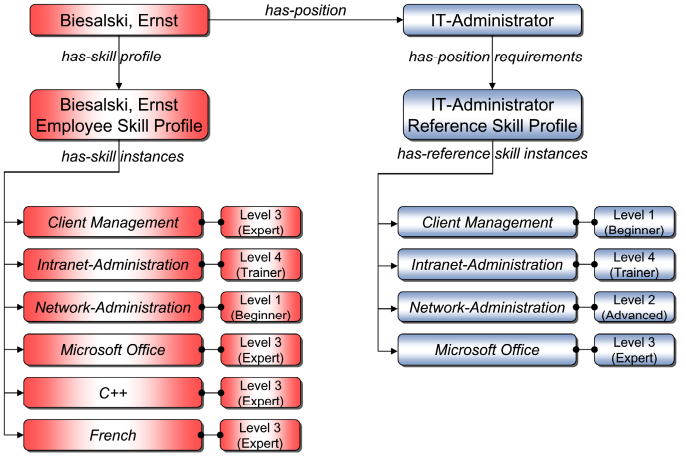


Fig. 3. Example: Part of Ontology-based Employee-Competence Database

In the same manner, in the module **"Detection of training needs / Training planning"**, we compare competence profile of jobs with the as-is profile of employee in order to identify qualification gaps. Such a comparison can also regard strategic directions, e.g. of the department, thus leading to a training planning and behavior which is closer to the company's overall strategic needs. When having identified strategic training gaps, they can be compared with training offers, and suggestions for filling gaps can be made, also regarding time restrictions of employees, budget restrictions, etc. Such suggestions can then be offered by the department manager to the employees. In the module **"Project and job staffing"**, the matching procedure searches best available employees for vacancies in project teams, based upon matching between to-be (project/ job requirements) and as-is (personal profile) comparisons.

3 The Module, "Project and Job Staffing"

In industrial practice, open jobs are seldom staffed along a structured procedure. Personal networks and preferences often play the major role. Of course, this guarantees neither a fast nor an optimized result, in particular when staffing a large project team with a number of heterogeneous required skills and competencies. Since large enterprise usually have electronic data about the competences and experience of their employees, a (partial) automation seems feasible and desirable.

In order to support project staffing, we need on one hand position skill requirements and on the other hand employees' skill profiles. We want to find – with a minimum staffing effort – the best suited employees who fulfill the position skill requirements as well as other constraints, such as availability. For integrating project staffing into a comprehensive HRM approach, we particularly need the standardized skill catalog which allows for a unified semantic description of position skill requirements and employee skill profiles, and a skill-matching procedure. For addressing these issues, we adopted the approach of [4], introduced a comprehensive ontology-based skill catalog at DaimlerChrysler plant Wörth / Rhine, as well as ontology-based similarity measures for profile matching.

Table 1. Some Central Concepts of the Skill Ontology

Concept	Description
Skill	The ability to produce solutions in some problem domain. In Enterprise Skill Management, the identified, named, trainable competence of some employee, often required to perform a specific organizational task (well), to fill a position, or to enact an organizational role.
Weight	Achieved or required level of expertise which indicates to which extent the given competence is or shall be mastered.
Skill instance	A skill together with a level of expertise (e.g., "expert in C++ programming", "beginner in English"). As a unit of reference only needed for internal technical reasons.
Skill Profile	A list of skill instances. As an *employee skill profile* describing as-is situation regarding an individual's personal qualifications, as *position skill requirements* describing necessary qualifications to do a job successfully.

3.1 Ontology-Based Skill Modelling

In general, a *skill catalogue* contains skills relevant for the company. In a *structured skill catalogue,* the skills can be equipped with a weight, describing the achieved level of expertise. In the DaimlerChrysler Wörth case, about 700 single skills have been modeled. For an employee, a personal *skill profile* lists his or her actual skills, together with weights for the achieved level of expertise. A profile for *position skill requirements* is a list of weighted skills which are important for successfully doing the required tasks in a given position (here the weight expresses the relevance of having achieved the specified level of expertise). Both profiles refer to the same, unified vocabulary specified in the skill catalogue (cp. Figure 2 and Table 1).

3.2 Project Staffing with Ontology-Based Similarity Measures

As an efficient and expressive data structure for processing skill profiles, we use an ontology-based approach [7] which stores, manages, and compares profiles with the help of the KAON [8] ontology management framework. For supporting the selection of qualified employees, our "project staffing" module is a web-based tool which allows to define project-specific position skill requirements and – based upon those – gives dedicated project staffing advice. To this end, decision-supporting information is taken into account from sources such as employee-skill profiles, job catalog, time recording system, etc. Since a multitude of perspectives must be fed into the complex employee selection process, the matching procedure which compares job profiles and potential candidates' profiles should be capable of semantically assessing the similarity of ontology instances. For realizing such a candidate selection procedure, we employed the similarity framework introduced in [9]. However, while [9] focuses on text-dominated application areas (comparison of vocabularies and terminologies), the comparison of skill profiles requires more advanced combination and expression means for similarity measures. So, we extended the framework such that the user can be provided with different metrics for assessing a candidates' suitability for a given job profile. Our requirements analysis and analysis of existing HRM systems showed that different similarity metrics for profile comparison should be used to express different aspects relevant for different tasks. For project staffing, an aggregated metric is used which combines the following four aspects:

- **Direct skill comparison:** we require an *exact match* of as-is and to-be. So we can specify *K.O. criteria* for the central requirements, especially in strategically important jobs.
- **Proportional similarity:** we identify also *partially fulfilled requirements*. This is also important if we can plan for additional teaching and qualification measures or for "training on the project".
- **Compensatory similarity:** we identify not only partially fulfilled requirements, but also *overqualifications*; so, additional expertise on one hand may compensate deficiencies on the other hand. If several employees fulfill the K.O. criteria, this can be useful to find the most suited one.
- **Taxonomic similarity:** the taxonomic structure of the skill ontology is taken into account to find "close matches" in the case that no employee has exactly the

required qualifications. Also usable for deciding between several candidates, and for refining profile specifications.

3.3 Similarity Measures for Skill Profiles

The basis of our skill-profile matching is the mapping of all competence metrics to a four-level scale (beginner, advanced, expert, teacher). Comparison of skill profiles is reduced to the comparison of skill instances. Let:

- R be a profile for some position-skill requirements consisting of a non-empty list of skill instances r with skill name rs and expertise level rl, and
- E be an employee-skill profile consisting of a list of skill instances e with skill name es and expertise level el

$$sim_{\text{skill name}}(r,e) := \begin{cases} 1, \text{if } \text{rs} = \text{es} \\ 0, \text{else} \end{cases} \qquad sim_{\text{skill level}}(r,e) := \begin{cases} 1, \text{if } \text{rl} = \text{el} \\ 0, \text{else} \end{cases}$$

3.3.1 Direct Skill Comparison

Often we want to specify special skill instances as K.O. criteria. This requires an extension of our skill modeling. Then we can define the direct skill comparison metrics for a position skill requirements profile R and an employee skill profile E as follows:

$$sim_{\text{direct comparison}}(R,E) := \frac{\sum_{r \in R, e \in E} weight(r) * sim_{skilllevel}(r,e) * sim_{skil\,\text{ln}\,ame}(r,e)}{\sum_{r \in R} weight(r)}$$

3.3.2 Proportional Similarity

The idea of direct comparison leads to the effect that each underfulfillment of a skill requirement finally results in a complete disqualification of the respective employee. It does not allow to assess the possibly differing extent to which the requirements where not fulfilled (only marginal deficiencies versus complete disqualification). In reality, project leaders need a metrics which is suited to assess partial fulfillment of requirements in an appropriate manner. To this end, we define the proportional similarity:

$$sim_{\text{proportional-similarity}}(R,E) := \frac{\sum_{r \in R, e \in E} weight(r) * sim_{proportional-level}(r,e) * sim_{\text{skill name}}(r,e)}{\sum_{r \in R} weight(r)}$$

with:

$$sim_{\text{proportional-level}}(r,e) := \begin{cases} 1 \text{ if } \text{rl} \le \text{el} \\ 1 - (\text{rl} - \text{el}) * \frac{1}{4}, \text{else} \end{cases}$$

This definition assumes that all the mutual "distances" between two subsequent skill levels are equal (thus it supposes sort of a "linear learning curve" on the way through different skill levels). It ignores ideas such as individual learning curves, increasing learning speed through continuously accumulated prior knowledge, etc. We took into account such models, but finally kept the pragmatic, linear approach.

3.3.3 Compensatory Similarity

Proportional similarity is an extension of the compensatory similarity which addresses not only under-, but also overqualifications. These can be especially useful when several candidates have fully satisfied the requirements of the other similarity measures and cannot yet further be distinguished.

$$sim_{\text{compensatory-similarity}}(R,E) := \frac{\sum_{r\in R, e\in E} weight(r) * sim_{compensatory-level}(r,e) * sim_{\text{skill name}}(r,e)}{\sum_{s\in S} weight(s)}$$

with:

$$sim_{\text{compensatory-level}}(r,e) := 1 - (\text{rl} - \text{el}) * \frac{1}{4}$$

This metrics must be interpreted differently from the other two presented before. In contrast to the situation with an exact match or a proportional similarity, a "1" is here not anymore an indicator that all requirements are fulfilled completely. Instead, overqualifications in some skill-profile facets may compensate for underqualifications in other facets.

3.3.4 Taxonomic Similarity

It is often difficult to find employees which fit relatively exact into a given profile specification. This comes also from the fact that it is not always trivial to specify the expected requirements unequivocally if there are different possible opinions how to characterize the required profile in terms of a complex skill catalogue which might provide many, slightly different, but related skills in a certain competence area. For instance, knowledge about "Spreadsheet software" might also be proven by a certificate about using "Microsoft Excel". Depending on the perspective, profile models may differ, both when employees describe themselves, and when project leaders define a required skill profile.

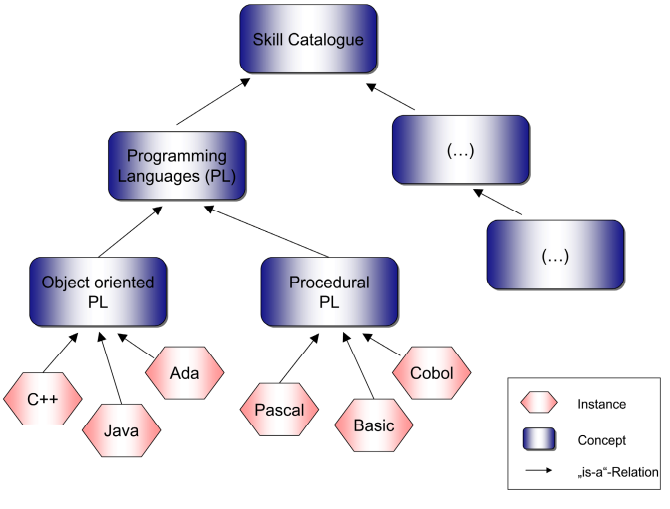


Fig. 4. Example: Excerpt of Skill Catalogue

Taxonomic similarity can be derived from semantic cotopy of two ontology instances. Figure 4 gives an example as a small excerpt from a hypothetical skill catalog: the skills "Object oriented programming" and "Procedural programming" are closely related since the have the same parent concept. While "Java programming" and "C++ programming" are very similar, "Pascal programming" is still related, but far more loosely. Such sophisticated comparisons of profiles based on the taxonomic skill catalog as background knowledge, allows far-reaching detailed assessments of whole project team staffs, if required. In particular, they allow for fine-granular ranking of candidates. Due to space limitations, we don't go into details about the computation of taxonomic similarity, here. For more information, see, e.g. [9].

4 Summary and Conclusions

The idea of detailed ontology-based modeling of personal skills is not new [6,10], but has found just recently more practical interest. Our own work within DaimlerChrysler, but also for instance [4,5,14,15] show that such an approach – if appropriately supported by organizational processes – can lead to more efficient and more effective project staffing in real-world, large-scale industrial application scenarios. As sketched in Section 2, even more impact can be achieved by designing a comprehensive ontology-based skill-management infrastructure, joining up existing systems, adding new functionalities, and designing suitable support processes (see also [2,3]). The focus of this paper was the use of ontology-based similarity measures for skill-profile matching. Many academic approaches for using *advanced reasoning* for skill matching (like [11]) apparently have not yet been applied in large-scale real-world scenarios. *Similarity-based approaches* seem not yet very widespread in this area, but they have proven their practical usefulness in a vast amount of Case-Based Reasoning (CBR) applications (see, e.g., [12]). CBR tools have also been successfully applied in Expert Finder systems which can be seen as a very specific partial instance of a skill management system [13]. The specific contribution of the work presented in this paper is to settle the CBR ideas upon a state-of-the art ontology infrastructure – thus combining the advantages of similarity based search (good retrieval results in vaguely specified query situations and complex domains) with those of ontology-based systems (clear semantics, good application potential for integration of different legacy systems).

When writing this paper, the software framework is fully specified, the HR Data Warehouse and the employee-skill database are already implemented, and the three application modules are under implementation.

References

1. Statistisches Bundesamt - Pressestelle Wiesbaden. Bevölkerung Deutschlands bis 2050 - Ergebnisse der 10. Koordinierten Bevölkerungsvorausberechnung. Statistisches Bundesamt, Wiesbaden (2003), In German
2. Biesalski, E., Abecker, A.: Ansätze zum ontologiebasierten Human Resource Management. In: WM2005. Workshop on IT Tools for Knowledge Management Systems: Applicability, Usability, and Benefits, Kaiserslautern (April 2005) (in german)

3. Biesalski, E., Abecker, A.: Integrated Processes and Tools for Personnel Development. In: 11th Int. Conf. On Concurrent Enterprising ICE-2005, Munich (June 2005)
4. Hefke, M., Stojanović, L.: An Ontology-Based Approach for Competence Bundling and Composition of ad-hoc Teams in an Organisation. In: Tochtermann, K., Maurer, H. (eds.) I-KNOW 2004, Graz, Austria (June 2004)
5. Lau, T., Sure, Y.: Introducing Ontology-based Skills Management at a large Insurance Company. In: Modellierung-2002. Modellierung in der Praxis - Modellierung für die Praxis, pp. 123–134 (2002)
6. Stader, J., Macintosh, A.: Capability Modeling and Knowledge Management. In: ES 1999. Applications and Innovations in Expert Systems VII, 19th Int. Conf. of the BCS Specialist Group on Knowledge-Based Systems and Applied Artificial Intelligence, pp. 33–50. Springer, Heidelberg (1999)
7. Staab, S., Studer, R.: Handbook on Ontologies. Springer, Heidelberg (2004)
8. Mädche, A., Motik, B., Stojanović, L.: Managing Multiple and Distributed Ontologies in the Semantic Web. VLDB Journal 12(4), 286–302 (2003)
9. Ehrig, M., Haase, P., Stojanović, N., Hefke, M.: Similarity for Ontologies - a Comprehensive Framework. In: PAKM 2004. Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability (2004)
10. Liao, M., Hinkelmann, K., Abecker, A., Sintek, M.: A Competence Knowledge Base System for the Organizational Memory. In: Puppe, F. (ed.) XPS 1999. LNCS (LNAI), vol. 1570, Springer, Heidelberg (1999)
11. Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., Mongiello, M., Mottola, M.: A Formal Approach to Ontology-Based Semantic Match of Skills Descriptions. J. Universal Computer Science 9(12), 1437–1454 (2003)
12. Watson, I.: Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufman Publishers, San Francisco (1997)
13. Vivacqua, A., Lieberman, H.: Agents to Assist in Finding Help. In: ACM Conf. on Computers and Human Interface (CHI-2000), The Hague, Netherlands (April 2000)
14. Dittmann, L.: Towards Ontology-based Skills Management. Project Report 8/2003, Project KOWIEN, University Duisburg-Essen (2003)
15. Hiermann, W., Höfferer, M.: Skills Management: Searching Highly Skilled Employees for Teambuilding and Project Management Tasks. In: Tochtermann, K., Maurer, H. (eds.) I-KNOW 2005, Graz / Austria (June 2005)

Part III

Information Systems Analysis and Specification

Comparing and Evaluating Ontology Construction in an Enterprise Context

Eva Blomqvist, Annika Öhgren, and Kurt Sandkuhl

Centre for Evolving IT in Networked Organisations (CENIT)
Department of Electronic and Computer Engineering
School of Engineering, Jönköping University
P.O. Box 1126, SE-551 11 Jönköping, Sweden
{blev,ohga,saku}@jth.hj.se

Abstract. Structuring enterprise information and supporting knowledge management is a growing application field for enterprise ontologies. Research work presented in this paper focuses on construction of enterprise ontologies. In an experiment, two methods were used in parallel when developing an ontology for a company in automotive supplier industries. One method is based on automatic ontology construction, the other method is a manual approach based on cookbook-like instructions. The paper compares and evaluates the methods and their results. For ontology evaluation, selected approaches were combined including both evaluation by ontology engineers and evaluation by domain experts. The main conclusion is that the compared methods have different strengths and an integration of both developed ontologies and used methods should be investigated.

1 Introduction

Ontologies have been used in many contexts and for many purposes throughout the years. Finally, through the interest in the Semantic Web and other recent research efforts, semantic technologies using formally represented ontologies have also started to find their way into to the business world. Thus, the area of ontology engineering has also been developing fast, new methods and tools are introduced continuously. When considering small-scale application cases within companies the need for reducing effort and expert requirements in ontology engineering is obvious. One way of reducing the effort is by further facilitating semi-automatic construction of ontologies, sometimes denoted Ontology Learning (OL), and use design patterns to reuse previous knowledge and experiences. Other ways are to further detail the existing manual ontology engineering methods, introducing reuse and more detailed guidelines.

Our earlier research has resulted in two different methods for constructing enterprise ontologies, a manual method described in [1] and a semi-automatic method, exploiting ontology patterns for the construction process, described in [2] and more recently further developed in [3]. These methods have been used in parallel during a project with industrial partners and a subsequent evaluation and comparison of the results was conducted. Enterprise ontologies in this case denotes domain and application ontologies within enterprises, for structuring of enterprise information.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 221–240, 2008.

Section 2 presents definitions and related work concerned with ontology engineering and ontology evaluation. Experiences from the project are presented in section 3, and the setup and results of the evaluation in section 4. In section 5 some recent developments in the project are discussed and the applications of the ontology are outlined. Finally in section 6 a discussion about the results is presented and some conclusions are drawn.

2 Background

This section presents background and definitions with respect to ontologies and ontology engineering, together with an overview of existing evaluation approaches.

2.1 Ontologies

An ontology can be described as a set of concepts describing a specific domain of knowledge, that can be used to create a knowledge base. Commonly the set of concepts can be partially structured into a hierarchy, like a taxonomy. An ontology thereby contains concepts, a subsumption hierarchy, arbitrary relations between concepts, and possibly other axioms (it may also contain other constraints and functions). At this time we do not restrict ourselves to a single ontology representation formalism, but for some parts of the construction methodologies a way to temporarily reduce the ontologies to a semantic net-like representation is assumed. In the experiments and evaluations described in this paper the ontologies have been represented in different frame-based languages (see [4] for an overview of ontology representation paradigms) but could also, with some effort, be transformed into for example OWL (see [5]) as discussed later in this paper.

Even using such an ontology definition, ontologies can be used for different purposes, and can be constructed in many different ways. One of the most common ways to describe the level of generality of an ontology is by using the structure suggested by [6], where a general top-level ontology can be specialised into a domain ontology or a task ontology. Domain and task ontologies can in turn be specialised, and combined, into application ontologies. Our main focus in this work is on domain and application ontologies within enterprises, hereafter denoted enterprise application ontologies. Another categorisation is to classify ontologies by their intended use, as in [7]. There are three main levels, terminological ontologies, information ontologies, and knowledge modelling ontologies, where each level adds further complexity. This work is concerned with enterprise ontologies intended for structuring and retrieval of information relevant to the enterprise in question.

2.2 Ontology Engineering

Classic ontology engineering has been a mainly manual process, assisted mostly through ontology editor tools, but leaving the actual modelling and knowledge acquisition completely up to the ontology engineer. Some manual methods are for example described in [4] and compared in [1]. Sometimes the construction process has been more an art than engineering discipline although recent methods have largely turned towards a more structured and well-defined process.

In addition ontology patterns have been suggested to aid the ontology construction process. There can be ontology patterns on several levels of abstraction, as noted in [8] and also in more recent publications like the NeOn project deliverable [9]. Ontology design patterns have been developed analogously to for example software design patterns, both for the logic parts of ontological modelling in different representation languages (like in OWL, see [10]) and for specific parts of the ontology content (as in [11] and [9]).

Recent research has also suggested to partly automate the ontology construction process, by for example assisting the knowledge acquisition process through automatic processing of enterprise internal documents etc. Semi-automatic ontology engineering, sometimes denoted Ontology Learning (OL) as in [12], [13] and [14] for example, has so far been largely focused on extracting single ontological primitives (concepts, relations, specific axioms, instances etc.) from document corpora. The output quality has also so far been quite poor, requiring a great deal of manual post processing. So far, to the best of our knowledge, no other semi-automatic methods have been proposed that exploit ontology patterns in a semi-automatic way.

2.3 Ontology Evaluation

Since ontologies are becoming more and more common there is also an urgent need for well-defined evaluation methods. This has resulted in a recent growing interest in ontology evaluation and numerous conferences and workshops devoted to the subject. The approaches that do exist have so far been quite differing, both in their aims and actual process. Some methods are used to determine how to choose between several ontologies, while others aim at validating a single ontology or correcting or preventing common modelling errors.

In [15] the authors describe an overall framework intended to cover all aspects of ontology evaluation and selection criteria, in order to give some structure to the field of ontology evaluation. The framework contains a meta-ontology describing the elements that may be evaluated and in addition an ontology of ontology evaluation and selection. The authors also try to structure the area of ontology evaluation into three kinds of evaluations, depending on the view of the ontology. If the ontology is considered as an information object, then structural evaluations can be applied based on the syntax and semantics of the ontology as it is represented. Next, if the ontology is considered as an information object *and* a language (the intended conceptualisation) functional evaluations can be applied, connected to the purpose of the ontology. Finally, the ontology can be treated from a usability viewpoint, where evaluations concern for example user satisfaction and efficiency of the ontology applications.

A deliverable by the Knowledge Web Consortium, see [16], also tries to give an overview of the state-of-the-art in ontology evaluation. They additionally identify three different stages of evaluation, namely evaluating an ontology in its pre-modelling stage, its modelling stage, or after its release. The first stage involves evaluating the material the ontology will be based on, the second stage checks the ontology correctness while building it, and the final stage involves comparing existing ontologies and monitoring ontologies in use. These stages can also be connected to the levels of evaluation suggested by [15], since some types of evaluations may be performed in only one stage of

the ontology development cycle while others may fit several stages, for example structural evaluations may be applied both during and after ontology construction.

Evaluation During Construction. The notion of continuous evaluation within the construction process is quite natural. Compare for example to software engineering where of course the code modules are tested and validated during construction, and not only evaluated as a whole at the end. When evaluating single ontologies during (or right after) construction, guidelines exist for among other things manually evaluating the correctness of the ontology. One approach is described in [17] and [4], where the focus is on evaluating a taxonomy. The guidelines are quite brief, so some expert knowledge concerning ontology engineering is definitely needed in order to perform the process. The idea is to spot and correct common development errors, like circularity in the taxonomy definitions or an instance belonging to two disjoint concepts.

Another approach is the OntoClean methodology, presented in [18] and also described in [4]. The methodology aims at exposing inappropriate or inconsistent modelling choices by using metaproperties to characterise the modelled knowledge. Three properties are discussed; rigidity, identity, and unity, and these can be used to evaluate if subsumption has been misused in the ontology. Rigidity for examples describes the philosophical notion of a property being essential to an instance, like the property of being a person is true at all times for a specific instance while the property of being a student might change from true to false in the lifetime of an instance. Using such properties to characterise the modelled concepts problems can be discovered in the ontology that would later lead to errors in reasoning services provided using the ontology.

Most manual ontology engineering methodologies also provide, more or less detailed, guidelines to assist the ontology engineer. These are perhaps not evaluation methods as such but could be viewed more as checklists or best practises in order to produce a better resulting ontology. Such guidelines may for example include naming conventions for primitives in the ontology and guidelines of module division and structuring of the taxonomy (consistent levels of detail etc.) and are part of most well-established methodologies (as noted in [4]).

Evaluation After Construction. In continued analogy to software engineering, it is not only important to evaluate the pieces during construction but also the ontology as a whole. This is both true for an ontology built from scratch as well as ontologies considered to be reused or integrated into a current project (ontology selection). A quite mature method dealing with comparing and selecting ontologies is the OntoMetric framework described in [19]. The method uses a multilevel framework of characteristics as a template for information on existing ontologies. Five dimensions are used; content, language, methodology, cost and tools. Each dimension has a set of factors which are in turn defined through a set of characteristics. The evaluation results in an overall score of the suitability of the ontology in a specific case. In order to tailor the method, a subset of the dimensions and factors can be used depending on what is deemed important in the specific project at hand, and the factors can also be ordered and have different weights. To ease the evaluation of ontology concepts glosses (natural language explanations) could be generated, as in [20], to let domain experts evaluate concepts without the aid of ontology experts.

A similar approach, using quality factors and an ontology of knowledge quality, is described in [21]. Here the focus is more on "objective quality" while in OntoMetric the focus is on subjective usefulness and selection based on the case at hand and the stakeholders involved. Yet another similar approach is presented in [22], where the authors suggest that the meta-models of the ontologies can be used to compare them. As previously mentioned the authors of [15] aim to unify most such evaluation frameworks in a formal way, in order to structure the complete field of ontology evaluation. Their meta-ontologies provide a similar framework describing available evaluation methods and evaluation subjects.

A very natural way of comparing and evaluating ontologies is of course to test how well they perform on certain tasks. Such an approach is suggested in [23] but it is based on a "gold-standard", which can be very hard to decide on and time-consuming to construct. With an enterprise focus this is usually not a feasible approach, since there is not one single correct way of modelling the enterprise. Also as noted in [24] there are, as of now, no standard tools for evaluating ontologies in specified task environments, so objective comparison of task performance does not seem to be possible at the moment.

When it comes to ontology content there exist different ways to compare the content similarity of two ontologies. Such approaches have also been implemented with the aim to match and integrate ontologies, like Chimaera described in [25] and PROMPT described in [26]. There are also others which for example measure cohesion of ontology concepts and modules, as in [27]. This last approach is simple but gives a good and intuitive idea of how the ontology is organised by computing average values of for example the number of taxonomic relations per concept etc. Also, in [15] several such measures, like breadth, depth, tangleness, fan-outness, cycle ratio and density, are suggested for structural evaluation. In line with these suggestions of course ontologies can be compared based on simple measures like size (number of concepts and relations) or metadata like author and construction date. If the ontology is constructed by an authority in the field this might mean more when choosing among ontologies than would some abstract measure of its correctness. In the position paper [28] the authors even suggest to establish an ontology certification authority, but this is still very much a vision of the future.

IR-Related Approaches. Measures that compare the ontology to the content it is supposed to represent are also discussed in [15], mainly as functional measures evaluating how well an ontology represents the given domain. In [24] the authors describe why classical Information Retrieval methods and measures, like precision and recall, cannot be used directly to evaluate ontologies or ontology construction methodologies in general (although other authors do use this for special cases, like in [20]). Instead [24] suggests an architecture for evaluating the fit of an ontology to a certain corpus of texts. This is done by extracting information, expanding the information and then mapping it against the ontology.

3 Experiment

This section describes the experiment performed in an enterprise context, to develop an ontology for the same purpose and with the same scope but using two different methods,

a manual and a semi-automatic method. The experiment was part of the research project SEMCO (Semantic Structuring of Components for Model-based Software Engineering of Dependable Systems). SEMCO aims at introducing semantic technologies into the development process of software-intensive electronic systems in order to improve efficiency when managing variants and versions of software artifacts. One concrete task supporting the overall aim would be to structure and annotate all documents produced throughout the development process, to maintain connections between initial requirements and their respective influence on specifications, and to possibly store prototypical parts of these documents in a domain repository for future retrieval and reuse.

At the time of the experiment the scope was to construct a selected part of the enterprise application ontology for one of the SEMCO project partners, a company within the automotive suppliers industry. The purpose of the ontology is to support capturing of relations between development processes, organisation structures, product structures, and artifacts within the software development process. The ontologies are so far limited to describing the requirements engineering process, requirements and specifications with connections to products and parts, organisational concepts and project artifacts, thus not the complete development process.

The two methods for constructing ontologies are quite new and have previous to this scenario only been used in smaller research test-cases. Both construction processes used the same set of project documents as starting point and major knowledge source. Furthermore, for the evaluation the same methods, tools, and evaluation teams were used.

3.1 Manual Construction

In a previous paper we have described the development of a methodology to fit the requirements in small-scale application contexts, see [1]. Below we give a short description of the proposed methodology consisting of four phases; requirements analysis, building, implementation, and evaluation and maintenance.

In the requirements analysis phase formalities of the ontology are specified, e.g. the the purpose and scope, intended users and uses etc. Usage scenarios of how the ontology can be applied should be developed. In order to shorten the development time, one step is to check whether there are any ontologies that can be integrated with the one being built. The building phase is iterative and uses a middle-out approach. The implementation phase primarily consists of implementing the ontology in an appropriate ontology editor tool. The implemented ontology finally needs to be evaluated to check that it fulfils the requirements. It should also be evaluated according to criteria such as clarity and reusability.

The manual construction in our project followed these four phases. First of all a user requirements document was produced. Information was mainly given by the SEMCO project leader, for example on intended users and uses of the ontology, purpose and scope. Different knowledge sources were identified, and we also looked for other ontologies to integrate but found none which was considered relevant. In the building phase the starting point was to use the available project documents as a basis for building a concept hierarchy. The method uses a middle-out approach, so first the most essential concepts are identified and then these are both specialised and generalised in

the hierarchy. It was decided that natural language descriptions for each concept were not necessary at this point. It was quite hard to derive relations, constraints, and axioms from the documents so after document analysis focus was switched to interviews with selected employees at the company.

The interviews were performed in two sessions, and where semi-structured. The interviewees were selected both based on their interest in the project but also based on having an overview of the complete development processes (although still being involved in the actual processes, not working on a managerial level). At the first session the interviewees discussed the top-level concepts (with assistance from the interview leader who had some knowledge of ontologies and ontology modelling), then went further in the hierarchy discussing each concept and its subconcepts. Feedback was given as suggestions, like "Restructure this" or "This concept is really not that important". After the first interview session the ontology was changed according to the suggestions. The second interview session was carried out similarly, resulting in minor corrections.

The evaluation and maintenance phase was partly integrated with the building phase, where the interviewees reviewed the ontology. The other parts of the evaluation are described in the following sections. The maintenance part has not yet been performed.

The resulting ontology has 8 concepts directly beneath the root and 224 concepts in total. In Fig. 1 a small part of this ontology is illustrated. Due to readability reasons, some parts of the structure are hidden in the illustration, still some of the main subject areas of the ontology can be noted. Processes are an important part of the ontology, and how processes connect to different organisation units and individual roles in the company. Additionally, since the focus is on the requirements engineering process, important concepts are requirements and product parts, and how the requirements

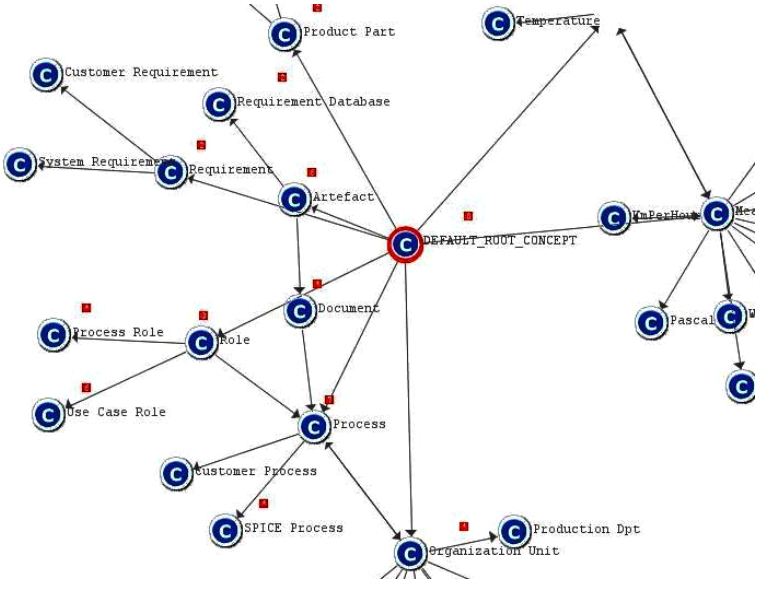


Fig. 1. Part of the ontology resulting from the manual construction

are represented within documents. Additionally concepts like, measurable qualities and units are included to be able to describe products and the semantics of the requirements.

3.2 Automatic Construction

As mentioned previously there are a number of existing semi-automatic approaches for ontology construction. In most cases there are existing knowledge sources, the question is how to extract the knowledge from these sources automatically and reformulate it into an ontology. Some of the parts present in most systems are term extraction by linguistical analysis and relations extraction by co-occurrence theory. Some systems also try to automatically build a concept taxonomy by using concept clustering or other methods, these are the ones that come closest to being fully automatic. Our method on the other hand aims on introducing the possibility of completely automatic construction, although we certainly acknowledge the need for user interaction in order to produce the best possible ontology output.

One important aspect when trying to reduce the effort in ontology construction is to also introduce reuse into the development process. The idea of using patterns is a commonly accepted way to reduce development effort and increase reuse in for example software engineering, but the ontology community has not yet adopted the idea on a broader scale. There exist a few pattern-based approaches for ontology construction as mentioned previously, but it is not yet common practise to include this in manual ontology construction methods nor in the semi-automatic approach that exist today.

Our approach, as presented in [2] and more recently in further detail in [3], uses design patterns as buildings blocks for ontology construction. Our approach also uses existing tools to extract concepts and relations. The general idea is to take the extracted terms and relations, match them against design patterns, and depending on the result use parts of the patterns to build the ontology. As described in [3] the method has now been further developed and extended, but for the experiment treated in this paper the method was mainly focused on a simple version of pattern selection and combination (as described in [2]).

In our research we adopt a very pragmatic view on ontology patterns. Mainly a pattern can be any kind of reoccurring object or sets of objects. In this sense patterns can also be modelling components based on some notion of best-practises, that is a pattern could be a partial solution as well as a more abstract template. Patterns can then be further divided based on their level of abstraction (as presented in [8]) and depending on their status as consensual template structures (usually denoted *ontology design patterns* in other literature) or simply as reusable components. For the rest of this paper we will not distinguish between the consensual ontology design patterns and the more pragmatic notion of reoccurring solutions.

For the experiment described in this paper 25 patterns were developed (examples can be found in [2]). The text corpus used consisted of software development plans and software development process descriptions. The matching of the patterns against the extracted concepts was done using a lexical matching tool. Next, the score representing matched concepts was weighted together with a score of matched relations into a total score for each pattern. This resulted in 14 patterns above the predefined threshold. The

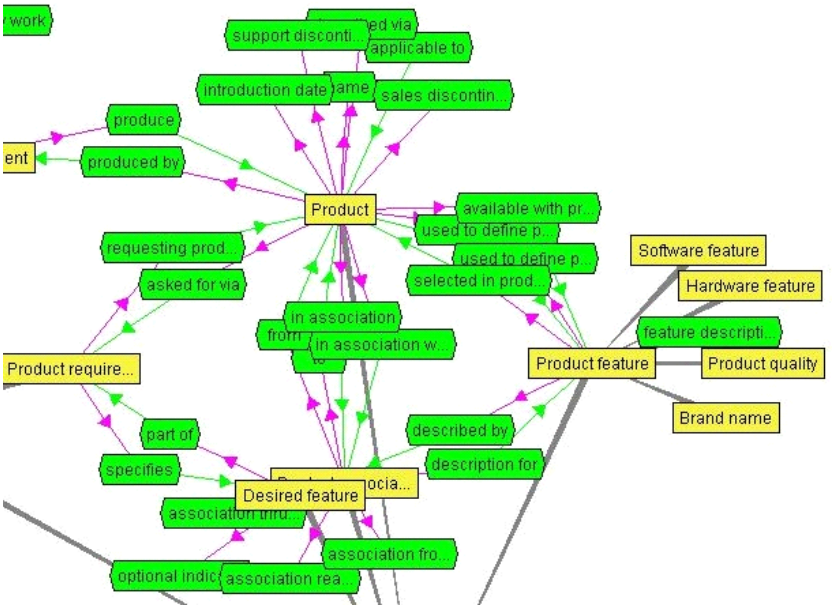


Fig. 2. Part of the ontology resulting from the automatic construction

accepted patterns were compiled into an ontology using some heuristics on the pattern combination.

The resulting ontology contains 35 concepts directly beneath the root concept and in total 85 concepts. In Fig. 2 a small part of this ontology is illustrated. Some parts of the ontology are hidden in the illustration, this is mainly due to readability reasons. The part shown in the illustration is devoted to describing the product concept, in connection with product features and product requirements. In the illustration the wealth of relations is easily visible. As mentioned the ontology has quite a number of concepts on the top level, this is due to that no general structure exists among the patterns, all the top concepts of the patterns end up at the top level of the ontology. Top concepts are for example both requirement, product and features.

4 Evaluation

This section presents the choice of evaluation methods for the resulting ontologies, and a description of the evaluation and its results.

4.1 Evaluation Setup

A decision was made to use several evaluation approaches, both intended for ontology expert and domain expert review, to get a broader view of the ontologies and indirectly also the construction methods.

First, a general comparison of the ontologies was needed to get an idea of differences and similarities. This comparison was done based on some intuitive metrics, like number of concepts, average number of attributes per concept, average number of subclasses per concept and average number of association per concept (as for example suggested in [15]). Also, the cohesion metrics from [27] were used, since we feel that they complement the other measures well. These metrics are: number of root classes, number of leaf classes and average depth of inheritance tree.

Second, an evaluation was performed by internal ontology experts using the two most well-known approaches for taxonomic evaluation, presented in [17] and [18]. Internal ontology experts were used for these evaluations, mainly because of their previous knowledge of the evaluation methods. Since we are evaluating both the ontologies and (indirectly) the methodologies for creating them, the idea was that the errors discovered could give valuable indications on advantages and disadvantages of each construction method.

Finally, to evaluate the content of the ontologies and their fit to the intended scope a subset of the OntoMetric framework in [19] was used. For our purpose only the dimension *content* was deemed interesting, and only one level of characteristics for each factor. Some characteristics were not applicable to both ontologies and since this is mainly a comparison, these were taken out of the framework. The computation of the final score was not performed, since the number of factors and characteristics were low enough to give a general impression. Domain experts from the company in question formed the evaluation team, but internal ontology experts prepared the material, assisted through the evaluation and collected the results.

The most desirable method of evaluation would of course be to apply the ontologies in their intended application context. This is not yet possible though, since the resulting application of the SEMCO-project is still being developed. Automatic gloss-generation could be a future improvement of the OntoMetric method, but at this time no such tool was available. Finally, the reason for not using any IR-related approach was mainly that this would give an unfair advantage to the semi-automatically created ontology since this was constructed using similar methods and completely based on the available company documents.

4.2 General Comparison

First some general characteristics of the ontologies were collected. In Table 1 these are presented for the two ontologies (the ontology constructed using the semi-automatic method is denoted “Aut” and the manually constructed ontology is denoted “Man”). The results show that the automatically created ontology has a large number of root concepts, it lacks some abstract general notions to keep the concepts together in groups, subject areas or views. It is also quite shallow and many concepts lack subconcepts altogether. Despite this, the concepts are much more strongly related through non-taxonomic relations and thereby has more attributes than in the other ontology.

The manually created ontology on the other hand contains a larger number of concepts. It also contains a top-level abstraction dividing the ontology into intuitive subject areas. There are few attributes and relations, this might be due to that many attributes are actually represented by other specific concepts, they are just not connected by an

Table 1. Comparison of general characteristics

Characteristic	Man	Aut
Number of concepts	224	85
Number of root concepts	8	35
Number of leaf concepts	180	64
Avg depth of inheritance	2,52	1,95
Avg number of rel. concepts	0,13	0,79
Avg number of attributes	0,01	0,46
Avg number of subclasses	1,00	0,57

appropriate relation. Relations seem to be harder to elicit from interviews than the concepts themselves.

4.3 Evaluation by Ontology Engineers

Two evaluation methods were used in the expert evaluation, first the general taxonomic evaluation criteria and then the OntoClean framework.

Taxonomy Evaluation. The ontologies were evaluated by ontology engineers according to the criteria presented in [17]. The criteria are the following:

- Inconsistency: circularity, partition and semantic errors
- Incompleteness: incomplete concept classification and partition errors
- Redundancy: grammatical redundancy, identical formal definitions of concepts or instances

There exist no circularity errors in the automatically created ontology since there is no multiple inheritance present, this also prevents most errors belonging to the inconsistency partition errors group. Multiple inheritance in the manually created ontology occurs only in a few cases, and no circularity errors were discovered among these. This also reduces the possibilities for partition errors, as mentioned previously. There are no exhaustive decompositions or partitions specified in either ontology so this eliminates the possibility of finding any other kind of partition errors.

Semantic inconsistency errors are more subtle to discover. This is a question of identifying wrong classifications. In the automatically created ontology there exist two concepts which could be thought of as wrongly classified since they make no sense in the context of this ontology, they are simply “junk” which happened to enter the ontology due to the immaturity of the ontology construction process. Semantic inconsistencies could also occur when two overlapping patterns are both included in the ontology, but this seems not to be the case in the ontology at hand. Concerning the manually created ontology these errors can only be assumed to have been discovered in the interview sessions with the domain experts, at this stage no more such errors were discovered.

Next, the incompleteness criteria was examined. Incomplete concept classifications might exist in the semi-automatically created ontology due to concepts missing in the patterns or in the text corpus used to develop the ontology. Since this will be an application ontology, not the whole domain needs to be modelled but only the parts needed

for this specific application. When comparing the two ontologies though, the semi-automatically created ontology seems to lack more specific concepts, such as names and company specific terms. This is definitely a problem originating from the semi-automatic process in itself, since it did not at this time contain any way to determine subsumption relations between input and patterns (only direct overlap). Even for the manually constructed ontology it is difficult to determine the incompleteness criteria until the ontology is used in its intended context, but in comparison to the semi-automatically constructed ontology it is much more complete on the lower levels of abstraction.

Several occurrences of partition errors were found in the semi-automatically constructed ontology, especially lack of disjointness definitions. This could be included in the patterns in order for it to propagate into the created ontologies, or more recent methods (like [29]) could be used to try to discover disjointness directly from the text input. Also some cases of exhaustive knowledge omission were found, but on the other hand the knowledge might not be needed for this specific application. In the manual construction process disjointness and exhaustive partitions were not discussed before building the ontology, so it is at this point not certain that there is a need for it. Deciding this ought to be part of the construction methodology, an important addition to the manual methodology.

Finally, there are no concepts with identical formal definitions but different names or redundant subclass-of relations in either ontology. Redundant subclass-of relations are not present in the patterns used in the automatic approach and no overlapping patterns have introduced it in this case. It is worth studying when considering overlap between patterns though, and methods to resolve such problems are needed in the semi-automatic ontology construction process.

OntoClean. Next, another expert evaluation was performed, this time by using the OntoClean methodology. Every concept in the ontology was annotated with the properties rigidity, identity and unity. This resulted in a backbone taxonomy containing 25 concepts in the semi-automatically constructed ontology. Here two violations of the unity and anti-unity rule were found and one violation of the incompatible identity rule. When analysed the unity problems arise because in this company "work" is seen as a "product", but "work" is generally not a whole. The identity conflict has the same cause since it is a question of "work" being defined as subsumed by the concept "product", but products in general are identified by a id-number while work is not. This is a quite serious problem which requires some consideration to solve in a good way, so that the solution still reflects the reality of the company in question but does not on the other hand introduce undesirable properties of the ontology when used in an application.

For the manually constructed ontology, the backbone taxonomy contains 178 concepts. One violation of the unity and anti-unity rule was found, and none of the other kinds of errors. The violation exists between the concepts "function" and "code", while a function is a clearly defined unit the concept of code is more abstract and cannot generally be seen as a homogeneous unit. This violation exists mainly due to that the abstraction level differs too much among the concepts on the same level of the taxonomic hierarchy of the ontology. The fact that no other violations were found is perhaps due to the simple structure of the ontology, it is very much like a simple taxonomy of terms.

Table 2. Result of the OntoClean evaluation

OntoClean rule	Man	Aut
Incompatible identity	No	1
Incompatible unity criteria	No	No
Unity/anti-unity conflict	1	2
Rigidity/anti-rigidity conflict	No	No

A summary of the results is presented in Table 2, where the manually created ontology is denoted "Man" and the semi-automatically created ontology is denoted by "Aut".

4.4 Evaluation by Domain Experts

The third step of the evaluation process include an evaluation performed by domain experts from the company in question. The evaluation was done based on a part of the OntoMetric framework in [19] as mentioned in the evaluation setup previously. Only the dimension "content" was considered and also no final score was computed, since the assessed characteristics are quite few and can tell us much about the nature of the ontologies in themselves. The evaluation was performed by a team of domain experts, working at the company in question, but the process was guided by an experienced ontology engineer in order to explain the notions to be evaluated to the evaluation team. For the evaluation a standard ontology editor user interface was used, where the ontologies were visible as graph structures. The ontology engineer was instructed to assist the team in understanding the general semantics of the ontology primitives and the functionality of the user interface but not to lead the evaluators by explaining the content of the ontologies (concepts and relations etc.).

The dimension "content" contains four factors: concepts, relations, taxonomy and axioms. For each of these factors characteristics applicable in this case were chosen. The scale suggested in [19] ranging from "very low" to "very high" in five steps was used as scoring, so for each evaluated issue the evaluation team members had to agree on a subjective score between "very low" and "very high". The characteristics used and the resulting scores for each ontology are presented in Table 3, where "Man" denotes the manually created ontology and "Aut" the semi-automatically created ontology.

As illustrated in Table 3 both ontologies seem to contain an appropriate number of concepts, neither is considered too small or too large and both seem to cover the intended scope, but the concepts in the manually constructed ontology are deemed more essential. This is most likely due to that the concepts are more specific. The semi-automatically created ontology also lacks some general abstract concepts to give it a comprehensible structure, which sometimes confuses the evaluation team since the ontology has no intuitive division into subject areas. On the other hand, the semi-automatically created ontology contains more attributes and relations (a higher density of each concept), which help to describe and define the concepts and reduce the need for natural language descriptions of each concept. The evaluators are faster to grasp the meaning of these concepts than some of the concepts in the manually constructed ontology that are only described through their placement in the taxonomy.

Table 3. Result of the domain expert evaluation

CHARACTERISTIC	SCORE				
	"Very low"	"Low"	"Medium"	"High"	"Very high"
CONCEPTS					
Essential concepts in superior levels	Aut			Man	
Essential concepts		Aut			Man
Formal spec. coincides with naming			Aut	Man	
Attributes describe concepts	Man			Aut	
Number of concepts				Man Aut	
RELATIONS					
Essential relations		Man		Aut	
Relations relate appropriate concepts			Man	Aut	
Formal spec. of rel. coincides with naming			Aut	Man	
Formal properties of relations	Man			Aut	
Number of relations		Man			Aut
TAXONOMY					
Several perspectives			Man Aut		
Maximum depth			Man	Aut	
Average number of subclasses			Aut	Man	
AXIOMS					
Axioms solve queries		Man	Aut		
Number of axioms	Man	Aut			

The semi-automatically created ontology contains many more non-taxonomic relations than the manually created one, even such relations that the company might not have thought of itself but which are still valid (this is explicitly noted by the evaluation team). The manually created ontology mostly contains relations explicitly stated by the company and easily expressed in words (either in documents or through the interviews). It is the non-taxonomic relations that give structure and comprehensiveness to the semi-automatically created ontology while the manually constructed ontology relies on specificity of concepts and precise naming (more company specific terms).

The semi-automatically created ontology of course also has a taxonomic structure, even though it lacks both some abstract top-level and the most specific levels compared to the manually created one. Despite this, it is perceived by the evaluation team as having quite a large depth, most likely due to the detailed division of the intermediate levels. This detailed division of the taxonomy is due to the detailed taxonomies present in the patterns used as a basis for the ontology. The manually created ontology has a larger number of subclasses per concept since a high number of very specific concepts exist, at a higher level of abstraction the average number of subclasses per concept is comparable between the ontologies.

The number of axioms is low in both ontologies, and the ones present are very simple. More advanced "business rules" is something that the company might need if the implemented application using the ontology is to function efficiently, especially if extended to handling more advanced use-cases than simple structuring and retrieval of documents and information within documents. In the manual method the question is how to elicit such rules using interviews, which is not a well-specified task so far. In the semi-automatic method these should be included in the patterns but then of course needs to be appropriately matched to the knowledge extracted, and included or not based on this matching information. So far we are not aware of any method for matching and comparing axioms, so this is probably a suitable task for future research.

At the end of the evaluation, in addition to the evaluation of the characteristics, an interview was conducted with the evaluation team members in order to see what parts might be completely missing. Natural language descriptions of concepts was one such item of discussion. For the task to be performed by the implemented ontology the interviewed domain experts thought this was not needed, since it was quite clear from the naming and context how a certain concept would be used. In a longer perspective though, for evolution and maintenance of the ontology, this would still be desirable, since concepts and their meanings can also evolve and change during an application's lifetime.

5 Ontology Combination and Project Scenarios

As a result of this evaluation the constructed ontologies were considered to complement each other well. The manually constructed one providing correct and precise terminology as well as abstract division of the scope into subject areas. The semi-automatically constructed ontology providing additional structure and a fine-grained definition of some of the most central concepts of the domain. These observations introduced the idea of subsequently combining the ontologies, in order to achieve the best possible result for the application case.

Since both of the constructed ontologies were built for the same case, and were partly constructed using the same knowledge sources they use approximately the same terminology. Due to this fact, no advanced ontology merging methodology was needed. Such methodologies usually assist in trying to unify two different sets of concepts using slightly different terminologies and possibly different scope. In our case it was fairly obvious which terms should be connected and which terms that were possible to merge into one concept. Based on this the merging process was performed completely manually. The process was started with an "empty" ontology, then the top-level concepts of the manually created ontology were added. The former top-level concepts of the automatically created ontology were thereby grouped as subconcepts of this structure. This step also resulted in some slight re-organisation of the top-level concepts, and the addition of some intermediate concepts, to make the two ontologies fit together and to get an even more intuitive structure. It was also considered important that all the siblings of a concept are on the same level of generality and that no new errors were introduced (with respect to the previous evaluations).

Next, the most specific concepts from the manually created ontology were inserted into the ontology, at the bottom level of the subsumption hierarchy. The fit between the two ontologies was not perfect, therefore some new intermediate concepts were introduced. This was due to the same reasons as the re-organisation at the top-level, to get an intuitive structure and to get all siblings on the same level of generality. All relations and attributes from the manually created ontology, which were not already in the automatically created ontology, were also included. This process resulted in an ontology with 379 concepts, where only 5 of them are placed directly beneath the root of the subsumption hierarchy. The intermediate "glue"-concepts which were added during the combination process amount to 18 percent of the total number of concepts.

As mentioned previously perhaps the most interesting evaluation is still to be performed, since the applications where the ontology will be used are still only partially developed. There are several applications of the ontology envisioned in the SEMCO-project, but so far only one of these have been implemented as an actual application. This scenario focuses on ontology-based artefact management, supporting reuse and comparison of artefacts between projects, as a part of the more general framework of a domain repository for the development processes.

To do this, an application for artefact management had to be developed. The main idea of the artefact manager tool is to use an enterprise ontology to define and store metadata and attributes of an artefact, as well as a link to the artefact itself. The enterprise ontology provides the attributes and the metadata, and artefacts are attached to it as instances, and connected to instantiated attributes. When artefacts have been stored in this way they can be searched, retrieved, and compared using their connection to the enterprise ontology. The ArtifactManager has been developed as a plug-in for the ontology development tool Protégé and has so far only been used by experienced engineers involved in the project, but not yet evaluated together with the intended users of the company. Thereby the final evaluation of the ontologies in this scenario is also yet to be performed. Still, it can be noted that the final merged ontology covers all the major parts concerning common artefacts in the requirements engineering phase, which leads us to believe that it can be used directly without any further configuration.

A second scenario envisioned in the project is the integration of feature models and enterprise ontologies, by supporting the feature metamodel in the enterprise ontology, with the aim of identifying similar requirements and product features in future projects. The features might be related also to the organisational elements in order to track responsibilities and expertise. In the end the aim is to try and generate internal requirements directly from detected features in the source documents (customer requirements), and the enterprise ontology and its feature model, based on semantic similarities between the source documents and stored requirements of previous projects. This scenario would require some additions to the current version of the enterprise ontology, and additionally an application supporting this scenario has still not been developed.

6 Conclusions and Future Work

With respect to the evaluation methods for performing this experiment some impressions can be noted: The general characteristics used give a good idea of the overall structure of each ontology, but they cannot be used directly to evaluate the ontologies. The intended scope and context of the ontology needs to be taken into account. If the ontology is solely intended for use by an automatic system there might not be a need for an intuitive top-level structure for example.

A similar problem applies to the taxonomic evaluation method used. This method might not be appropriate when the exact usage of the application of the ontology is still not specified in great detail. This is an explanation for the somewhat inconclusive results produced by that part of the evaluation. Thereby our conclusion is that although the method is sometimes presented as suitable both during ontology development and after completion of the ontology, during ontology and application development it can

only be seen as a set of guidelines. It can be used as an actual evaluation method only when all the requirements of the application to use the ontology are stable and the ontology is completed.

The OntoClean evaluation on the other hand produces more conclusive and exact results, which do not only expose faults in the ontology design but can point out difficulties and ambiguities of the real-world case at hand as we have seen. This evaluation produced valuable results and gave good insights in the nature of the concepts used in the ontology. The main problem is that it is quite difficult to perform and really requires a deep understanding of the metaproperties involved (it requires the knowledge of an experienced ontology engineer), additionally it requires deep knowledge of the actual semantics of the concepts and how they have been modelled in the ontology (it requires the knowledge of a domain expert). A conclusion is that although this is sometimes presented as an evaluation method that can be performed by ontology engineers, it requires in addition the deep understanding of a domain expert, which implies that it will either be very hard to perform or it has to be performed in a collaborative environment including both kinds of expertise.

The OntoMetric-framework produced some good results, but still has some disadvantages. Here the problem was to present the ontologies to the domain experts in an understandable way, without at the same time influencing the reviewers. This difficulty will always remain when using domain experts without ontology expertise, but perhaps it could be reduced in the future by using something like the gloss generation discussed in section 2.3. Also the more recent framework presented in [15] might, when detailed further, provide a better overall view of ontology evaluation aspects and methods.

Finally, the most interesting evaluation approach, where the ontologies are tested against their goals and application scenarios, is still not performed. This was not possible to include in this study since development of a pilot application in the SEMCO-project is still ongoing. Also no general task-oriented application environments exist for testing ontologies, as noted in section 2.3. To further explore the advantages and disadvantages of the two methods, some additional measures, like construction time, could be interesting to evaluate. This has not yet been possible, since the systems of the semi-automatic method are not yet fully integrated and thereby for example requires some file conversions and manual procedures which are not really part of the method.

To summarise the performed evaluations, some strengths and weaknesses can be noted in both the manual and the automatic approach. The automatic approach will probably never directly capture company specific names of concepts, since these will not be part of general patterns. Instead this has to be handled through utilising some external knowledge source and also including indirect coverage in the matching procedure. Also, the method can only capture what is in the patterns on the upper levels, so a choice has to be made whether to include some abstract concepts at the top level of most patterns or to be satisfied with a less coherent ontology. A third option would be to apply some general architecture patterns or possibly even reference architectures for enterprise ontologies, in order to introduce such a general structure. Finally, the correctness of the resulting ontology is very much dependent on the correctness of the patterns and the combination procedure. The automatic approach has its strengths in relying on well-proven solutions and easily including complex relations and axioms.

The manual approach gives a result with less complex relations and axioms, since the bottleneck in this case is the precise elicitation of these pieces of information from domain experts. Knowledge acquisition is an relatively old research field but still methods lack the ability to extract complex knowledge structures from people in an efficient way. Furthermore, the extent to which the application domain is covered by the ontology depends significantly on the interviewed experts, different domain experts might often present different views and might not agree on the state-of-affairs in the company. On the other hand, the manual approach has one big advantage, since it also captures the most specific concepts (and their precise names) that the enterprise actually uses. Also, the more abstract concepts at the upper level give an intuitive idea of the scope of the ontology, and divides it into views and subject areas.

Neither of the approaches produce many formal errors in the ontology, according to the evaluations, but some improvements can be made in both methods. Improvements of the automatic method could be to evaluate the patterns more thoroughly and improve the pattern combination method. The patterns could also be enriched with more axioms and natural language descriptions. Improvements of the manual approach could be to use a larger set of knowledge acquisition methods to elicit more complex structures from the document sources and domain experts.

The main conclusion that can be drawn is that the ontology engineering approaches each have both strengths and weaknesses and complement each other well, as was also noted in section 5 through the combination of the resulting ontologies. This might suggest that a combination of the development approaches could on the long term give the best results, but it is too early to state this firmly, since the methods have only been tested in parallel for one single case. The next step is to repeat this experiment in other cases in order to be able to generalise these results and perhaps arrive at some solution for combining the approaches.

Acknowledgements

This work is part of the research project Semantic Structuring of Components for Model-based Software Engineering of Dependable Systems (SEMCO) based on a grant from the Swedish KK-Foundation (grant 2003/0241). We thank all evaluation teams for contributing to the results of this study. The article is an extended version of a paper presented at the ICEIS2006 conference, special thanks also to the ICEIS conference reviewers for a valuable discussion on how to improve this paper.

References

1. Öhgren, A., Sandkuhl, K.: Towards a Methodology for Ontology Development in Small and Medium-Sized Enterprises. In: IADIS Conference on Applied Computing, Algarve, Portugal (February 2005)
2. Blomqvist, E.: Fully Automatic Construction of Enterprise Ontologies Using Design Patterns: Initial Method and First Experiences. In: ODBASE. Proc. of The 4th Intl Conf. on Ontologies, DataBases, and Applications of Semantics, Cyprus (2005)

3. Blomqvist, E.: Ontocase - a pattern-based ontology construction approach. In: Proccedings of OTM 2007: ODBASE - The 6th International Conference on Ontologies, DataBases, and Applications of Semantics, Vilamoura, Algarve, Portugal (to appear)
4. Gómez-Pérez, A., Fernández-Liópez, M., Corcho, O.: Ontological Engineering. Springer, Heidelberg (2004)
5. Web Ontology Language: Owl web ontology language overview. (February 2004), Available at: http://www.w3.org/TR/owl-features/
6. Guarino, N.: Formal Ontology and Information Systems. In: Proceedings of FOIS 1998, pp. 3–15 (1998)
7. van Heijst, G., Schreiber, A.T., Wielinga, B.J.: Using explicit ontologies for KBS development. Intl. Journal of Human-Computer Studies 46(2-3) (February 1997)
8. Blomqvist, E., Sandkuhl, K.: Patterns in Ontology Engineering: Classification of Ontology Patterns. In: Proc. of ICEIS 2005 7th International Conference on Enterprise Information systems, Miami Beach, Florida (May 24-28, 2005)
9. Suárez-Figueroa, M.C., et al.: D 5.1.1 neon modelling components. (March 2007), Available at: http://www.neon-project.org/
10. W3C-SWBPD: Semantic Web Best Practices and Deployment Working Group (2004), Available at: http://www.w3.org/2001/sw/BestPractices/
11. Gangemi, A.: Ontology Design Patterns for Semantic Web Content. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 262–276. Springer, Heidelberg (2005)
12. Maedche, A.: Ontology Learning for the Semantic Web. Kluwer Academic Publishers, Norwell (2003)
13. Buitelaar, P., Cimiano, P., Magnini, B. (eds.): Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press, Amsterdam (2005)
14. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, New York (2006)
15. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Qood grid: A metaontology-based framework for ontology evaluation and selection. In: Vrandecic, D., Suarez, M.C., Gangemi, A., Sure, Y. (eds.) Proceedings of Evaluation of Ontologies for the Web, 4th International EON Workshop, Located at the 15th International World Wide Web Conference WWW 2006 (2006)
16. Hartmann, J., Spyns, P., Giboin, A., Maynard, D., Cuel, R., Suarez-Figueroa, M.C., Sure, Y.: D1.2.3 methods for ontology evaluation. Version 1.3.1, Downloaded 2005-05-10 (2005), Available at: http://knowledgeweb.semanticweb.org/
17. Gómez-Pérez, A.: Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases. In: Banff Knowledge Acquisition for Knowledge-Based Systems, KAW 1999, vol. 2 (October 16-21, 1999)
18. Guarino, N., Welty, C.: Evaluating Ontological Decisions with OntoClean. Communications of the ACM 45(2), 61–65 (2002)
19. Lozano-Tello, A., Gómez-Pérez, A.: ONTOMETRIC: A Method to Choose the Appropriate Ontology. Journal of Database Management 15(2) (April-June 2004)
20. Navigli, R., Velardi, P., Cucchiarelli, A., Neri, F.: Automatic Ontology Learning: Supporting a Per-Concept Evaluation by Domain Experts. In: Workshop on Ontology Learning and Population (ECAI 2004), Valencia, Spain (2004)
21. Supekar, K., Patel, C., Lee, Y.: Characterizing Quality of Knowledge on Semantic Web. In: FLAIRS-2004. Proceedings of AAAI Florida AI Research Symposium, Miami Beach, Florida (May 17-19, 2004)
22. Davies, I., Green, P., Milton, S., Rosemann, M.: Using Meta Models for the Comparison of Ontologies. In: EMMSAD 2003. Proc. of Eval. of Modeling Methods in Systems Analysis and Design Workshop (2003)

23. Porzel, R., Malaka, R.: A Task-based Approach for ontology Evaluation. In: Workshop on Ontology Learning and Population (ECAI 2004) (2004)
24. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data Driven Ontology Evaluation. In: Proc. of Intl Conf. on Language Resources and Evaluation, Portugal (2004)
25. McGuinness, D., Fikes, R., Rice, J., Wilder, S.: An Environment for Merging and Testing Large Ontologies. In: KR 2000. Proc. of the 7th Intl Conf. on Principles of Knowledge Representation and Reasoning, Colorado, USA (2000)
26. Noy, N.F., Musen, M.A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: 17th National Conf. on AI (AAAI-2000), Austin, Texas (2000)
27. Yao, H., Orme, A.M., Etzkorn, L.: Cohesion Metrics for Ontology Design and Application. Journal of Computer Science 1(1), 107–113 (2005)
28. Orbst, L., Hughes, T., Ray, S.: Prospects and possibilities for ontology evaluation: The view from ncor. In: Proceedings of Evaluation of Ontologies for the Web, 4th International EON Workshop, Located at the 15th International World Wide Web Conference WWW 2006 (2006)
29. Voelker, J., Vrandecic, D., Sure, Y., Hotho, A.: Learning disjointness. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 175–189. Springer, Heidelberg (2007)

Modelling and Visualisation for Interoperability Requirements Elicitation and Validation

Sobah Abbas Petersen[1], Frank Lillehagen[2], and Maria Anastasiou[3]

[1] Norwegian University of Science & Technology, Trondheim Norway
sap@idi.ntnu.no
[2] AKM AS, Lysaker, Norway
f.lillehagen@akmodeling.com
[3] Intracom SA, Greece
mana@intracom.gr

Abstract. This paper describes a methodology and a model-based approach for supporting the requirements elicitation and validation work in the ATHENA project. Numerous interoperability requirements have been gathered by four industrial partners and these requirements are validated against interoperability issues. The process of obtaining requirements from industrial users and developing solutions for them involves several communities such as the users, stakeholders and developers. A model-based methodology and approach are proposed to support the analysis of the requirements and for incorporating the different perspectives and views that are desired by everyone. An example from the telecommunications sector is used to illustrate the methodology and a matrix-based validation approach is supported using a model developed in the Metis modelling environment.

1 Introduction

Advances in technology have facilitated the use of technology in all aspects of life, from business to health care, from education to manufacturing as well as in our everyday lives. The role of ICT and communication are becoming increasing significant in our lives. Computers and systems no longer operate as single, isolated bits of technology used by a single operator. Rather, the trend has been for one system to communicate with another or depend on input from another and for people and businesses to share information and collaborate.

We live in a diverse world and this diversity is no doubt reflected in the technology that we use. We often find ourselves trying to transfer data across heterogeneous systems, attempting to get two incompatible devices to communicate or wondering how our business partners' concepts and terminology map to ours. Standardisation efforts have helped address some of these issues. However, there is still a long way to go before we are able to collaborate with our partners without facing interoperability problems.

Interoperability, in particular, technical interoperability is not a new issue. However, focus on areas such as e-business, e-government, e-health and e-learning has created a greater interest in interoperability. This is evident from the updated eEurope

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 241–253, 2008.

2005 Action Plan where there is an emphasis on increasing interoperability in all these areas [1]. Interoperability has been recognised as fundamental to achieving Australia's e-government aims [2] and the National Institute of Standards and Technology in USA has estimated the cost of inadequate interoperability in some industries to be as much as $15 billion per year [3]. Thus, there is a global awareness on the significance of interoperability and a need to increase interoperability for improved business collaboration. One approach to address interoperability and produce solutions for interoperability problems is by identifying and analysing interoperability requirements that are posed by industry.

Two projects that have focused on interoperability are EU Integrated Project 507849 ATHENA (Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application) [4] and IST-508011 INTEROP Network of Excellence [5]. Both these projects conduct research on interoperability for networked enterprises. The INTEROP project focuses on theoretical research while the ATHENA project considers interoperability in industry by analysing the interoperability requirements from four different industry sectors, telecommunications, aerospace, automotive and furniture, and by developing solutions for interoperability.

The process of obtaining requirements from industrial users and developing solutions for them involves several communities such as the users, stakeholders and developers. The analysis of the requirements also involves several communities and numerous discussions. It is often difficult to keep track of the stages in this process and to take care of the knowledge that is created in this process that adds to the value of the solutions. One of the problems that have been identified during this process is fostering understanding among the different communities that are involved [6]. The facilitation of this process in itself poses interoperability problems! The requirements elicitation and validation processes are often seen in isolation by the solution developers and the views of the industrial user or the stakeholder are often overlooked. There is a need to consider the lifecycle of the requirement as a whole and take into account the views of the various communities that are involved in the different stages in the lifecycle.

Visualisation has long been recognized as a powerful technique for the analysis and understanding of requirements among the different communities [7]. The main focus of the work done in requirements visualisation has been in representing user requirements in different ways such as pictures or schematic diagrams using the concepts familiar to the user. 3D visualisation of requirements have been used so that the requirements can be viewed from different view points and not just analysed based on their measurements, e.g. [8] and [9]. These types of visualisations are beneficial for analysing the requirements of physical entities. Support for analysing and understanding requirements for business analysts is provided by supporting diagramming techniques such as those supported in Microsoft Visio, e.g. [10]. However, additional means of visualisation are required to analyse other types of requirements such as interoperability requirements.

This paper is based on research conducted in both the ATHENA and INTEROP projects. Interoperability is a property of an enterprise and it can be achieved either by re-engineering and reconciliation or by design. Active Knowledge Modelling (AKM) takes the approach that interoperability can be achieved by design and is based on the recognition that only work-centric knowledge truly integrates any enterprise and that

this knowledge must be expressed and used in an operational knowledge architecture to achieve full interoperability, predictability, traceability and sustainability. We propose the RAIS methodology and AKM for eliciting and validating interoperability requirements. The RAIS methodology takes into account the user and the stakeholders' views as well as the solution developer's view. The AKM approach facilitates modelling and inter-relating the different views and visualising them from different perspectives [11]. The approach described in this paper provides a flexible way of analysing a large number of requirements (interoperability as well as other types) using model-based visualisation techniques. Modelling facilitates the analysis of the requirements in the context of the enterprise and not in isolation. For example, an overview of the user that provided the requirement, the alternative solutions to meet the requirement as well as the organisational unit that will develop the solution and their resource situation can easily be viewed from a model. This facilitates an overview of any set of requirements in different contexts and a quick analysis and assessment of the situation. The modelling approach proposed also allows the porting of the requirements and related information to a web interface or to other viewing tools such as Protégé [12]. This is not a new method for requirements elicitation or validation. Rather, it is a complementary approach where existing requirements elicitation or validation methods can be used. This approach can then be used to provide visual support for the other methods.

The rest of this paper is structured as follows: Section 2 describes the ATHENA project and interoperability requirements; Section 3 describes the RAIS methodology and the model for analysing and validating the interoperability requirements; Section 4 illustrates the methodology and the model with the help of an example and Section 5 discusses the advantages of this approach and our directions for continuing this work in the future.

2 ATHENA Interoperability Requirements

The ATHENA project defines interoperability as seamless business interaction across organisational boundaries. It distinguishes between technical interoperability and business interoperability. Research into technical interoperability is conducted by Action Line A projects while the Action Line B projects conduct research on business interoperability by analysing scenarios from four industry sectors; aerospace, automotive, furniture and telecommunications, see Fig. 1. ATHENA emphasises on the

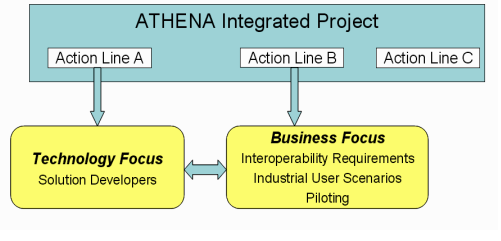


Fig. 1. ATHENA Overview

mutual dependence of the technical and business aspects of interoperability in producing good solutions. In addition to providing interoperability solutions, one of the activities of Action line A projects has been to identify interoperability issues (also referred to as issues) or problems concerned with interoperability which are used to validate against the requirements provided by the industrial partners in the Action Line B projects.

The ATHENA B4 project deals with interoperability requirements and one of the main tasks is to provide a way to easily validate the interoperability requirements by analysing them against the interoperability issues and the solutions.

2.1 Requirements Elicitation and Validation

Interoperability requirements from four industry sectors have provided a rich and diverse set of requirements. These requirements were derived by analysing different business scenarios, e.g. supply-chain management from the automotive industry and project portfolio management (PPM) from telecommunications. One of the main tasks that are currently being undertaken is the identification of requirements that are common to all these industries, similarities and differences in the requirements from the different sectors and using this information in the design of solutions.

A mapping approach has been defined for the validation of the requirements and solution against the interoperability issues to ensure that all the issues that have been identified have been addressed. This mapping approach also considers weighting to rank the impact of a particular issue on a requirement and the relevance of a solution to an issue.

Some important criteria in requirements validation that have been taken into account are:

1 Ensure that all requirements and interoperability issues are considered.
2 Ensure that all requirements and interoperability issues have proposed solutions.
3 Facilitate the analysis of the above two points, e.g. by supporting matrices to do this.
4 Ensure that requirements can be represented, viewed and analysed in different points of views and interests. e.g. the stakeholders' view or the users' view.

The large number of requirements that have been provided by the industrial users (~450) and managing and analysing them have been a challenge. The requirements are formulated in natural language and sorting or searching through them or identifying relationships among the requirements demands sophisticated techniques and technological support. The model-based approach supports the management of the large number of requirements and their relationships to the other aspects such as interoperability issues and solutions.

3 RAIS Methodology and Requirements Model

The RAIS methodology is described in Fig. 2, where the different concepts that relate to the requirements and interoperability issues and how they relate (or influence) are illustrated [13] and [14].

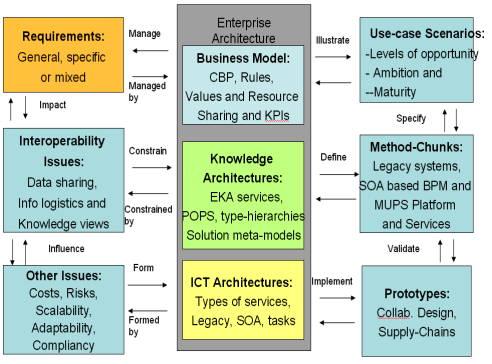


Fig. 2. RAIS Methodology

RAIS – Requirements, Architecture, Interoperability issues and Solutions, which are the main concepts of RAIS, bring together all the aspects of requirements engineering and analysis for the design and development of appropriate solutions. The main components are:

Requirements: These are interoperability requirements obtained from the industrial users.

Architecture: This is about the structure of entities, either systems or enterprises, their components, and how the components fit and work together to fulfill some purpose.

Interoperability Issue: These are problems concerning interoperability extracted and elicited from analysis of business scenarios.

Solution: These are the solutions that are designed by the ATHENA project as well as appropriate solutions that are available today.

These four components help us to address the what-how dimensions of a system; e.g. *what* is desired and *how* the desire is achieved [15]. In addition to these, it is possible to incorporate other aspects such as *who* desire the functionality, i.e. the stakeholders' view, or *where* in the business process this is relevant, i.e. the business and enterprise architecture view. By bringing these components together in a cohesive methodology, we are able to see the dependencies among these concepts and how they influence and impact one another. This can be done by modelling the dependencies among these different concepts. We have used AKM technology and the Metis modelling environment [16].

The RAIS methodology uses an operational Active Knowledge Architecture (AKA) to build up solid collaboration, learning and holistic design approaches to products, processes and systems. By using AKAs as the multi-dimensional integrator and interpreter of meaning, the concepts can be easily extended to cover experiences, new ideas, lessons-learned and specifications for improvement. Knowledge modelled using the RAIS methodology and an AKA architecture can readily be transformed into operational solutions and workplaces for engineers.

3.1 Modelling Concepts

In Metis, the notion of a metamodel is used to define the elements of a model. The main components of the RAIS methodology, requirement, architecture, interoperability issue and solution are represented as an entity-relationship model, see Fig. 3, where the different components are represented as objects and can be related to one another. The relationships between the different objects are obtained by adapting the RAIS methodology to the mapping approach described in the project [17].

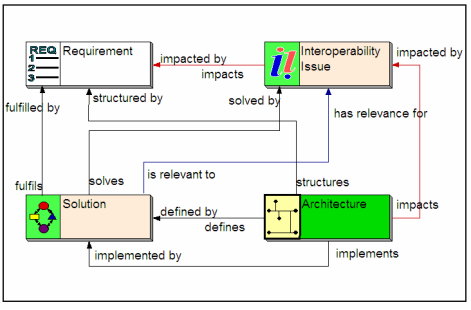


Fig. 3. RAIS Metamodel

The main concepts and relationships are:

- An Interoperability issue *impacts* a Requirement.
- A Solution *fulfils* a Requirement.
- A Solution *solves* an Interoperability issue.
- A Solution *is relevant to* an Interoperability issue.
- An Architecture *structures* Requirements.
- An Architecture *impacts* an Interoperability issue.
- An Architecture *defines* a Solution.
- An Architecture *implements* a Solution.

3.2 Requirements Model

A Metis model of all the interoperability requirements is available from the ATHENA Dynamic Requirements Definition System (DRDS), [18]. The DRDS has a web-based front end for the user to provide the requirements and a database that could be used to generate a model in the Metis modelling environment for requirements elicitation, validation and visualisation.

In Metis, a requirement is represented as an instance of the object type requirement. We have enriched this model by modelling the interoperability issues and solutions and creating relationships between them to indicate correlations.

Weighting of correlations are implemented by defining a property on the relationship that indicates the impact of a correlation. For example, the impact of an interoperability issue may be low (=1 or yellow), medium (=2 or orange) or high (=3 or red), see Fig. 4.

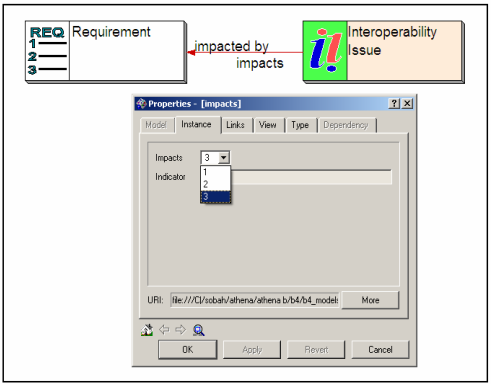


Fig. 4. Weighting on Relationships

The mapping approach defined for validation of requirements uses matrices to analyse correlations of requirements and issues and solutions and issues. The model supports automatic generation of these matrices where a relationship between two objects or sets of objects (which represent the two axes of the matrix) is marked on the corresponding cell on the matrix. We have used numerical values as well as colour coding to support the visualisation of this.

4 Example

In this paper, we focus on the interoperability issues identified by the telecoms sector by Intracom S.A., Greece and the interoperability requirements provided by them. Some of these interoperability issues are:

T4. Provision of (near) real-time aggregated views of key business information.
T7a. Legacy applications integration and interoperability
T7b. Model driven generation of interoperable custom and role-based workplaces
T8a. Communication / collaboration infrastructure integration / interoperability
T8b. Exchanged and/or shared data integration / interoperability
T8c. Distributed data and data access synchronization.

While these issues have been identified by the telecoms sector, they are not confined to this particular industry sector alone. Some of these issues, such as "T7b, model driven generation of interoperable custom and role-based workplaces", are likely to be issues that are relevant to other industry sectors as well.

The requirements and the interoperability issues from this sector have been modelled and the correlations between them have been established. A screen shot of this model is shown in Fig. 5.

4.1 Validation Matrices

A matrix of the requirements against interoperability issues is shown in Fig. 6 and Fig. 7. (Note that although identifiers of requirements and interoperability issues have

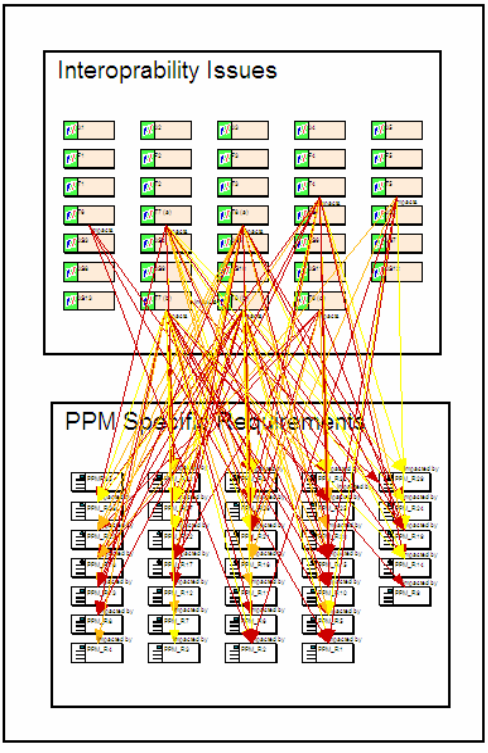


Fig. 5. Requirements and Interoperability Issues

been displayed on the matrices in the figures, it is also possible to display their names and descriptions.) The matrix shown in Fig. 6 shows the correlations between the requirements and interoperability issues and the colour code used to indicate the level of the impact is shown in the cell corresponding to each correlation. This is for quick, visual assessment of the impact of issues on requirements. The level of impact can also be shown as a quantitative value (Fig. 7) or as a qualitative value (low, medium, high).

By observing the matrix, it is possible to have an overview of the requirements–issues landscape. A correlation indicates that there is an impact. The values or the colours on the matrix indicate the level of the impact. And most importantly, it will indicate if there are no requirements that address a specific issue or vice versa:

- An interoperability issue that does not have a relationship to a requirement or does not impact any requirement indicates that new requirements must be considered so that this issue is addressed and will be considered in the development of solutions.
- A requirement that is not impacted by an interoperability issue indicates that it must be verified if this requirement is really an interoperability requirement.

Matrices can also be generated for the other elements in the model. For example, a matrix can be generated to validate the solutions against interoperability issues and to assess the relevance of each solution for an interoperability issue. The solutions can

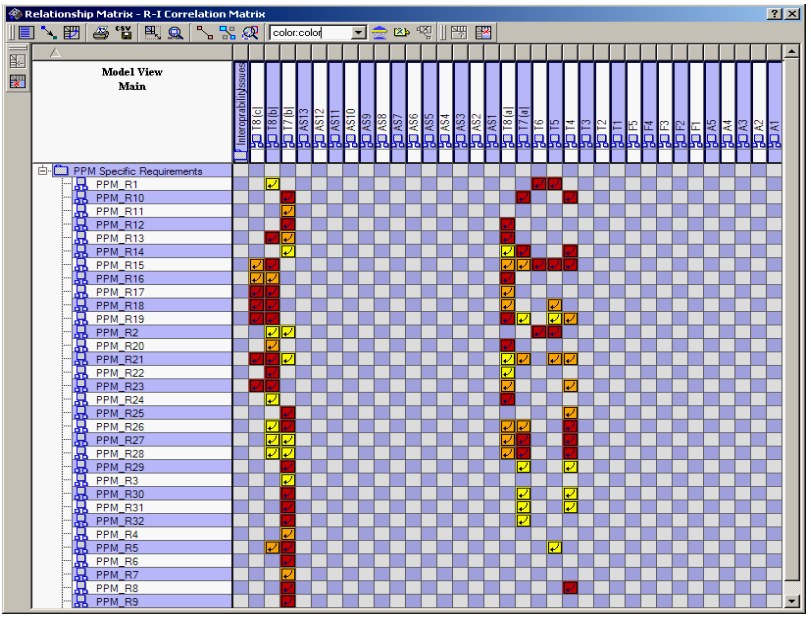


Fig. 6. Requirements and Interoperability Issues: Correlation Matrix

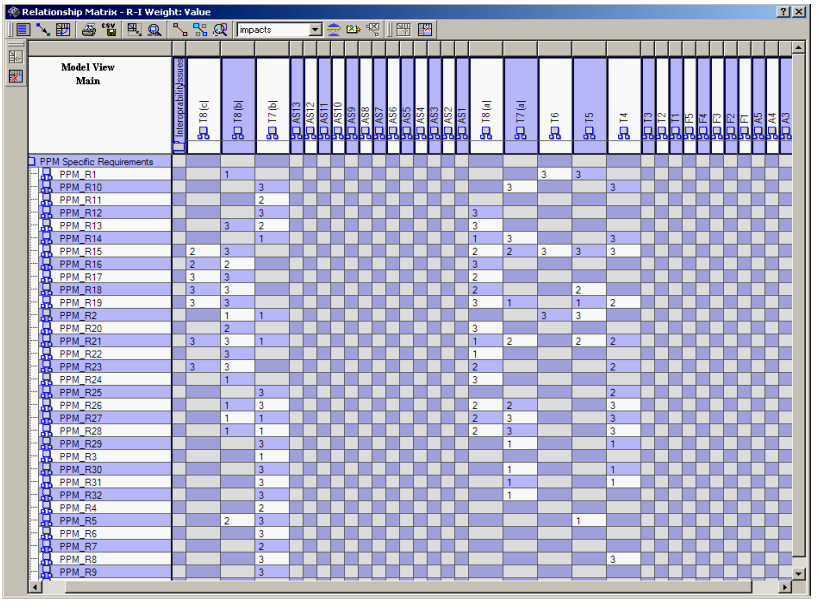


Fig. 7. Impact of Interoperability Issues on Requirements

be existing solutions, based on state of the art, or new solutions developed by the ATHENA project. A matrix of existing solutions against interoperability issues will identify problem areas where innovative new solutions can be proposed by ATHENA.

Similarly, a matrix containing both existing and new solutions can be used to identify solutions developed by ATHENA where the solution may be an improvement or an alternative to an existing solution.

4.2 Selective Viewing

One of the advantages in using a visual modelling environment is the possibility to do selective viewing of the data such as selecting one interoperability issue and viewing the requirements or solutions that are related to this issue. For example, the issue "T7b, model driven generation of interoperable custom and role-based workplaces" impacts several requirements. Similarly, several of the solutions developed in the ATHENA project address this interoperability issue. A selective view of this generated from the model, where the solutions for an interoperability issue and the different project workpackages that are responsible for developing the solutions are shown in Fig. 8. A matrix of a part of this view, interoperability issue and the requirements that are impacted by that issue can also be generated and this is shown in Fig. 9.

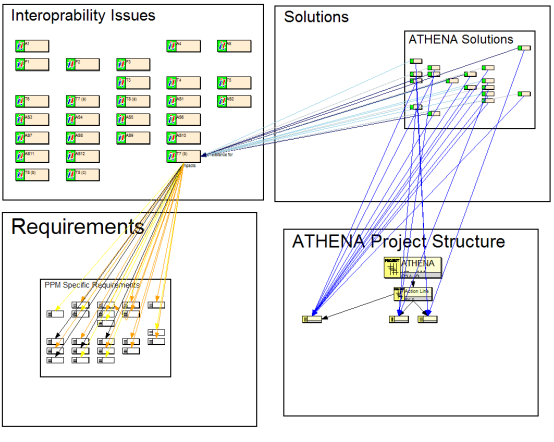


Fig. 8. Requirements impacted by one Interoperability Issue

This capability is particularly important when there are several communities involved in the work. For example, the industrial users are interested in identifying the issues and ensuring that there are requirements addressing all these issues. Solution developers are interested in ensuring that they provide solutions to relevant issues as well as meet the requirements from the industrial users. The stakeholders are interested in seeing the benefit that is achieved by adopting a particular solution. For example, in a situation where there are two alternative solutions that meet their needs, they will select the one that is most beneficial for them.

4.3 Editing the Model

One of the activities during the elicitation and validation process is changing or updating the information in the model. For example, we might want to have additional

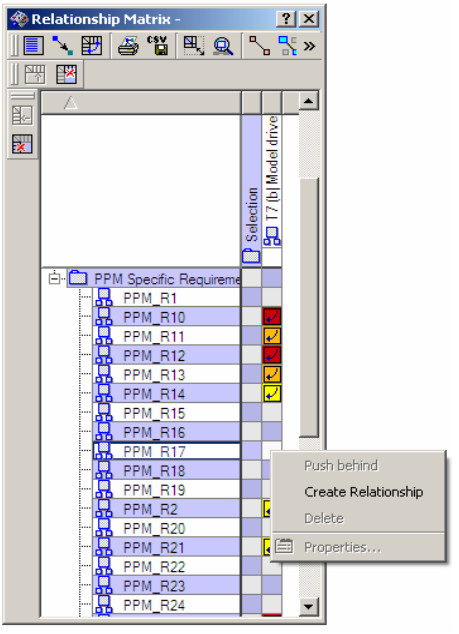


Fig. 9. Matrix to show impact of one Interoperability Issue

correlations, delete a correlation or change the value of the impact. The matrices can be used to change the correlations or edit the relationships as shown in Fig. 9. It is easier and more efficient to conduct an analysis on a selective view of the information and then use the menus available on the matrices to make changes. For example, we might want to have a correlation between the requirement PPM17 and interoperability issue T7b. Similarly, we might want to change the value of impact on a particular correlation. These changes can be easily achieved using the matrix, e.g. establish a new correlation between a requirement and an interoperability issue, delete an existing correlation and change the value of the impact of the correlation.

5 Discussion and Future Work

This paper describes a methodology and a model-based approach for supporting the requirements elicitation and validation work in the ATHENA project. Numerous interoperability requirements have been gathered by four industrial partners and these requirements are validated against interoperability issues. The analysis of these requirements supports the design and development of solutions. The use of matrices has been identified as a means to support the validation of requirements.

AKM is pursuing designing for interoperability and thus the approach facilitates easy viewing of the relevant concepts and provides enhanced visualising capabilities such as automatically generated matrices, selective views and colour coding on relationships to indicate a level or a degree of an impact or relevance. The model supports easy extension of the concepts as well as easy integration of work done in the other

parts of the project. It also supports easy and efficient changing or updating of the model contents during the validation work.

The main benefits of using this modelling approach is that it facilitates the visualisation of the interoperability requirements in relation to other aspects such as the solutions and project organisation and not just by themselves. The colour coding and numerical values used to indicate the degree or the level of the impact of a relationship provides a quick visual overview of the situation. This approach does not require extensive computation and allows the viewing of requirements via a web interface and by porting the information to other viewing tools.

We are currently enhancing our model with requirements and interoperability issues for the other industrial users in the project and mapping the solutions that have been developed in the ATHENA project against the interoperability issues. We plan to extend the model by adding new concepts such as the classification structure of the requirements which will further support the elicitation process and the identification of common requirements among the different industries. Another important view that we plan to implement is that of the stakeholder and the business value. This is particularly important in the design and validation of solutions, which is the next phase of our work.

In the future, we see these interoperability requirements, issues and solutions utilised by industry as well as other sources as a means of quickly assessing their interoperability problem(s) and finding or designing solutions in a fast and efficient manner.

Acknowledgements

This work has been carried out as part of the ATHENA B4 project. ATHENA Integrated Project is funded by the European Commission under the FP6 IST Programme. The authors would like to thank Dag Karlsen for his support in the development of the modelling template and the members of the B4 project, in particular Dario Leo and Giorgio Sobrito, and the ATHENA consortium for the interesting discussions that have inspired this work.

References

1. COM, eEurope 2005: An Information Society for All, Commission of the European Communities (2002)
2. Australian-Government, Technical Interoperability Framework, Version 2 (2005)
3. Gallaher, M.P., O'Connor, A.C., Dettbarn Jr., J.L., Gilday, L.T.: Cost Analysis of Inadequate Interoperability Analysis in the U.S. Capital Facilities Industry, NIST, NIST CGR 04-867 (2004)
4. ATHENA, Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application, http://www.athena-ip.org/
5. INTEROP, The European Virtual Laboratory for Enterprise Interoperability, http://interop-vlab.eu/
6. Christel, M., Kang, K.: Issues in Requirements Elicitation, CMU/SEI-92-TR-012 (1992)

7. Özcan, M.B., Morrey, I.C.: A Visual Requirements Validation Environment for the Reverse Engineering of Formal Specifications from Rapid Prototypes. Software Engineering Notes 20(5), 83–87 (1995)
8. Teyseyre, A.: A 3D Visualization Approach to Validate Requirements. In: Congreso Argentino de Ciencias de la Computación (2002)
9. Özcan, M.B., Parry, P.W., Morrey, I.C., Siddiqi, J.: Requirements Validation based on the Visualisation of Executable Formal Specifications. In: 22nd International Computer Software and Application Conference (1998)
10. stpsoft-Limited, Requirements Tools for Software Development, http://www.stpsoft.co.uk/index.html
11. Lillehagen, F.: The Foundation of the AKM Technology. In: Concurrent Engineering: Enhanced Interoperable Systems, Balkema, The Netherlands, pp. 700–715 (2003)
12. Protégé, http://protege.stanford.edu/
13. ATHENA, ATHENA Mapping Approach Validation Report, Version 0.6, ATHENA WD B4.6.2 (2005)
14. INTEROP, State of the Art: Exploration of Methods and Method Engineering Approaches, Version 1, Interop DTG6.1 (2005)
15. Soderberg, N.R., Crawley, E.F., Dori, D.: System Function and Architecture: OPM-Based definitions and operational Templates. Communications of the ACM 46(10), 67–78 (2003)
16. Metis, http://www.computas.com/templates/Page____371.aspx
17. ATHENA, ATHENA Mapping Approach: Definition between Requirements and Interoperability Issues, Version 2.0, ATHENA WD B4.6.1 (2005)
18. Solheim, H., Lillehagen, F., Petersen, S.A., Jørgensen, H.: Anastasiou: Model Driven Visual Requirements Engineering. In: 13th IEEE Requirements Engineering Conference, Paris, France (2005)

An Eclipse-Based Framework for Supporting Software Development Cooperative Activities

Arnaud Lewandowski and Grégory Bourguin

Laboratoire d'Informatique du Littoral, 50 rue Ferdinand Buisson
62228 Calais Cedex, France
{lewandowski,bourguin}@lil.univ-littoral.fr

Abstract. Software development is a cooperative activity, since it implies many actors. We focus on CSCW integrated global environments. Many studies have already shown, for a long time, that a 'good' cooperative environment should be able to take into account the users' emergent needs, and should be adaptable. Of course, such properties should also be found in environments supporting software development. However, our study of some existing platforms points out their lacks in terms of tailorability and cooperative support. Eclipse is one of these broadly used platforms. But even if it presents some shortcomings, its underlying framework offers some features particularly interesting for our purpose. Upon results previously obtained in the CSCW field, we propose to extend the Eclipse platform, in order to offer a new support for software development by creating a cooperative context for the activities supported in Eclipse by each integrated plug-in.

Keywords: Software development support, CSCW, tailorability, inter-activities, Eclipse.

1 Introduction

Following the continuous growth of information technologies, users are looking for systems able to support their intrinsically cooperative activities. And today, these activities tend towards being realized through complex systems supporting this cooperation of actors, distributed through space and time.

The software development (SD) domain, which provides such tools as well as uses them, does not derogate from this rule. Systems are more and more complex, their development requires the cooperation of many actors, with different roles and cultures. Many studies of common practices in SD show how this cooperative dimension holds a strong place in this field [15,19]. Actually, the necessity to take into account this dimension in SD environments has been underlined for a long time [13]. However, the current systems do not bring to the fore, or according to recent work on CSCW, poorly support the cooperative dimension of these activities.

We have been working for several years on the problems tied to the creation of global and integrated CSCW environments. Our work is inspired by results coming from the Social and Human Sciences (SHS), especially the Activity Theory (AT), and

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 254–268, 2008.

aims at proposing tailorable models and systems according to the expansive properties of every human activity. These thoughts led us to define the Coevolution principle [5].

This paper presents a proposition to better take into account the cooperative dimension in SD tools. Our approach proposes an extension of the Eclipse platform to integrate a cooperative dimension in accordance with the Coevolution principle. The first part of this paper presents the implications tied to the support of SD cooperative activities, by integrating the results of our work in the CSCW domain. Then we present the solution we propose, a cooperative extension to the Eclipse platform.

2 Cooperative Software Development

Software development environments generally provide a poor support to the cooperative dimension of this activity. From our point of view, adding a cooperative dimension does not simply consist in adding specific communication tools that will bring a new coloration in the environment; rather, it seems more valuable to integrate in a more fine-grained way such tools in the environment, but it raises also many questions. We have been working since years in the CSCW field, and we have to integrate as many results we obtained in this field as possible, if we want our cooperative SD environment to be a 'good' CSCW environment too. We present now the underlying elements of our work in this domain.

2.1 CSCW, Tailorability, Coevolution

For years, CSCW research has identified the need for tailorability in the systems. This necessity has been brought to the fore by many empirical and theoretical results, based on theories coming from the Social and Human Sciences, like Situated Action [22], Ethnomethodology [10], or Activity Theory (AT) [2]. Besides, our research is founded on AT, which has been broadly used in the domain over the last ten years [14, 18].

The AT gives information that can help systems designers to better understand the human activities they try to support. We cannot explain here all the results we obtained by founding our research on this theory, neither detail the reasons of our choices. More information can be found in [4]. However, in order to facilitate the understanding of the paper, we briefly remind some basic elements of the AT on which we based our reasoning.

We use the basic structure of an activity proposed by [11]. This structure presents the human activity as an interdependent system involving a subject that realizes the object of the activity, and the community of subjects who are concerned with this realization. Relations between the subject, the object and the community are mediated. In particular, the subject uses tools to realize the object of his activity. Rules determine what means belonging to the community, and a division of labor describes how the work is shared up by the members of the community. Furthermore, this structure as a whole is dynamic and continually evolves during the realization of the activity. For example, subjects may transform the mediating elements, such as tools, as new needs emerge. Subjects themselves evolve during the activity, acquiring skills and developing some experience of its realization. Thus, when subjects transform the elements participating in their activity, the experience they acquired crystallizes in

these elements. This experience, 'written' in the transformed artifacts, becomes available for others, which can benefit from it in other activities.

From our point of view, a system that supports a specific activity is a mediator of it. The system does not contain the activity, but rather takes part in it. A particular system supports a particular activity and this is why we call it an Activity Support (AS). Inspired by the AT, we have defined a set of concepts representing the elements participating in an AS. These concepts are represented in Figure 1. In an AS, a set of subjects are part of a community and realize a task. As defined by Leont'ev and taken up by Bedny and Meister in [2], a task is "a situation requiring the realization of a goal in specific conditions". The realization of the task corresponds to the activity supported by the system. Then, in our approach, the task specifies the object and a set of tools and roles involved in the activity. The role represents a part of the division of labor and some rules existing in the activity supported by the system. It defines how a subject may use the tools allowing it to act in the system. As it has been underlined by Christiansen [8], the tool enables and limits the actions that may be performed by a subject. This fact is even more true with computer tools that may implement a part of the rules guiding the activity. A task may specify another task. For example, a task may need the realization of another task by other subjects to be completed. This link is useful to represent networks of activities as it has been proposed by Kuutti in [14]. Finally, when a task specifies another one, a role in the task may imply another role for a subject in the other task.

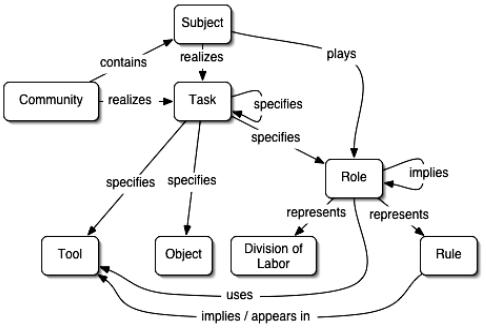


Fig. 1. The elements participating in an Activity support

These elements help us to define an AS. Inspired by the AT, we know (as we underlined before) that all these elements should evolve during the realization of the activity. Therefore, the AS has to be tailorable: it must provide the means to adapt it during the activity it is involved in. This is why we define the AS as a reflective system, as explained in Figure 2. In our approach, the task that is supported by the system is made available from the AS. This is realized by introducing particular tools we call meta-tools. The meta-tools allow the subjects to access to the activity definition, i.e. the task specification. In our approach, an AS is considered as the instance of a task. The task can then be inspected and/or transformed from the AS. We define a causal relationship between the task and the AS: transformations of the task have direct repercussions on the system. For example, modifying a role defined in the task directly affects the way the subjects playing this role will perform their activity in the corresponding AS.

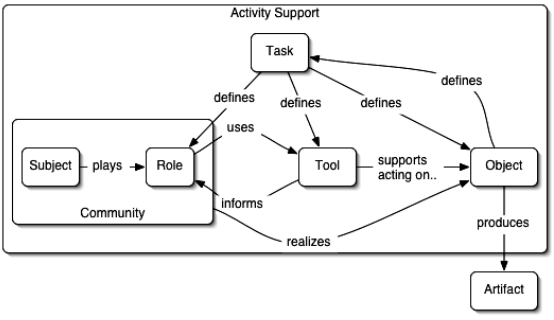


Fig. 2. An Activity Support and its task: a reflective system

This model underlines that, according to the mechanisms described in the AT, a part of the human activity is a meta-activity whose object is to reflect on the activity itself for solving contradictions that may appear in its constituting elements. In our AS model, using the meta-tools correspond to a meta-activity. As the meta-tools are managed in the AS as any other tool, the role of the subjects will affect the way they participate in the meta-activity too and then, the meta-activity is a real cooperative activity that is also supported by the system. This is a simple definition of our approach of tailorability that we have called the Co-evolution [5]: the system supports a particular cooperative activity like the development of a particular software, but it also supports its own cooperative (meta-)activity of (re)design.

Finally, this tailorability can be associated with some mechanisms of experience crystallization and reuse. These mechanisms are synthesized in figure 3.

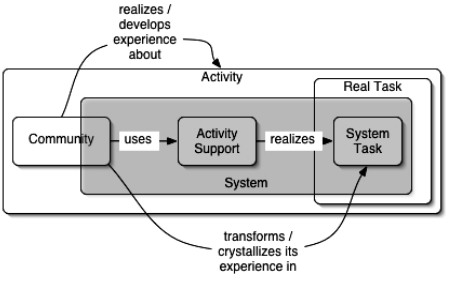


Fig. 3. Experience crystallization through the system and in the system task

The community realizes an activity in the real world. This activity is supported by the system. The system task is the part of the real task that has been specified inside the system to create an AS. The community acquires some experience while performing its activity. This experience can be made explicit through the system by making its task evolve. For example, an evolution in the division of labor in the real activity may result in a new set of roles specified in the system task and that will affect the corresponding AS. This new evolved task corresponds to a new AS model that can also be instantiated for another community that needs a computer tool support for realizing a similar task. The crystallized experience developed while the realization of

a task can then be transmitted through the tailorable system thus supporting an important mechanisms underlined in AT.

2.2 Shortcomings of Existing SDEs

Today, many platforms support SD cooperative activities. Actually, many studies have already been conducted on the evaluation of SD tools and environments. For example, Barthelmess and Anderson [1] focus on process-centered SD environments (PCSDE) found in the literature, analyzing and evaluating them from an AT viewpoint. What emerges from that study is that even if presenting positive aspects, "PCSDEs suffer from the production-oriented philosophy of software engineering", assuming that the routine steps in a process model are sufficient for the execution of an activity. Consequently, they observe a lack of tailorability in such systems. This observation is relevant not only for PCSDEs, as we will see by examining other environments. In this part, we have chosen to particularly focus on web portals and Integrated Development Environments (IDE) that are widely used by communities of developers during the SD process, in order to point out some of their general drawbacks.

Web portals, like SourceForge (http://source forge.net) and Freshmeat (http:// freshmeat.net), provide a global environment that integrates many tools, such as planning tools, concurrent versions systems, forums, bug reporting tools, etc. These integrated tools aim at supporting some cooperative activities tied to the SD process. A positive aspect is that these portals mostly take into account the major elements constituting an AS, as we defined them before. Especially, we find in such solutions some mechanisms that define, for example, the role of each community member (by granting them rights on the integrated tools). However, those web environments present some drawbacks, especially with regards to tailorability. Indeed, the latter is in most cases greatly reduced, since the available tools are defined *a priori* in the system. The dynamic integration of new tools is generally not possible. And when it is, this integration remains at a graphical interface level, which actually does not differ very much from using such tools outside the environment. In their study of such Collaborative Development Environments (CDE), or virtual spaces on the web, Booch and Brown [3] recognize that "there are a number of substantial barriers to successful adoption of a CDE", especially because "no CDE supports all the features" that should be found in an ideal environment. As we underlined before, we think that it is not possible to conceive such an 'ideal' AS *a priori*. Instead, a better solution – even if not easy – is to provide a tailorable environment able to be adapted to the needs emerging during the activity. The last important point we emphasized is that the artifacts constituting the AS should crystallize the experience of the subjects. It is especially true for the tools used in SD activities. For instance, developing a web site in php and developing a j2ee application – even if both are SD activities – may imply different development methods, tools, and even roles. The experience acquired during each activity may differ from one to another (methods, tools, etc.). An environment supporting the SD global activities should be able to crystallize these kinds of experience that could be useful in other similar projects. However, this feature seems to be missing in the web-based solutions examined, where it is not possible to reuse the experience developed during a project in another one.

Integrated Development Environments (IDEs), such as NetBeans or Eclipse, also integrate sets of tools dedicated to support producing code activities. Unfortunately, most IDEs only focus on these producing code activities, and avoid or forsake their cooperative dimension. As underlined by Sarma [20], "coding has traditionally been considered the most important activity of a developer in software engineering. As a result, tool builders have focus on creating better programming languages and environments that facilitate coding, while ignoring other activities". Therefore, the many elements identified that constitute an AS are mostly missing from such environments. Actually, IDEs provide gates towards a common repository – such as CVS (Concurrent Versions System) – that supports and manages documents sharing, but not the communication between developers. Some collaborative extensions to Eclipse try then to palliate this lack [7]. But from our point of view, even if needed, this kind of extensions – that provides some collaborative functionalities – still remains superficial, and does not tend to take into account the cooperation at a global level. Eclipse has not been designed in that orientation, and it does not manage any notion of role, or something like this that takes into account the status of a user in the global cooperative activity he participates in. As a result, the user has to integrate the tools (plug-ins) he needs himself, and to configure them according to his role in the real supported activity. Despite this drawback, some of these environments provide functionalities that foster tailorability: for example, Eclipse provides a powerful extension mechanism that allows the platform to be adapted (by integrating new plug-ins) to support new needs. In most environments, like in NetBeans and Eclipse, experience crystallization is reduced to patterns (supporting the creation of different projects) that will configure the environment in a way that seems suitable for such a project. Eclipse gives to the user other means to customize their environments: it manages 'perspectives' that are specific views, or visual arrangements of the tools in the environment. People can create their own perspectives that can be reused later, in similar projects. This feature could be improved by taking into account the cooperative dimension that constitutes an AS.

As we see, lacks remain in the existing global environments supporting some SD practices. Even if the many kinds of environment present interesting features and mechanisms, we note that none of them seems to meet the main 'requirements' of an AS we have identified. In practice [24], due to these lacks in commonly used platforms, the actors of SD use in a complementary way many tools or environments (IDE, PCSDE, web portals, synchronous discussion tool, etc.), each one supporting one or more (sub-) activities. Faced with such statements, we aim at proposing an AS that palliates these needs: a tailorable platform supporting SD cooperative activities, inspired by our previous work on the Coevolution principle.

3 CooLDev: Cooperation under Eclipse

The CooLDev project (Cooperative Layer for software Development) is directly inspired by results obtained during the DARE project that evolved until becoming CooLDA (Cooperative Layer supporting Distributed Activities) [4], the generic underlying platform on which CooLDev lays.

3.1 The Inter-activities Approach

A major choice in our reasoning to design a CSCW environment is to consider that many tools already exist, which are useful in supporting some activities we are interested in. Thus, our main goal is not to create such tools, like a new code editor. Rather, we want to create an environment that integrates these many tools. From our point of view, detailed in [6], each tool supports one kind of activity. When several tools are used in parallel by a group of actors, they generally serve a more global activity than the original activity they were designed for. For example, a group may use in parallel an IRC, a CVS, and a code editor. Each of these tools supports a particular activity (discussion for the IRC, etc.) but they do not know each other. However, they are used in a complementary way by the group since they are used in the context of a global cooperative activity: software development. In such a case, the coherence of the environment is mainly mentally managed by the users. Then, our purpose is to provide an environment that can create a context for the use of the different tools involved in a global cooperative activity (e.g. a SD activity), managing the links between its different (sub-)activities. Assuming that each tool supports a specific activity, our environment is intended to manage what we call the inter-activities.

To achieve this, we have created an activity model (Fig. 4) [6] – inspired by the elements presented before – that conceptualizes the elements constituting an AS and that allows the specification of the links of the inter-activities.

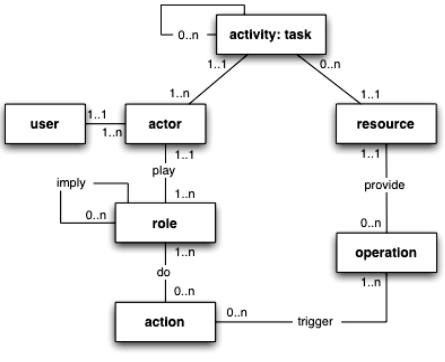


Fig. 4. The activity model of CooLDA

Each activity is linked to a resource that proposes operations. A resource corresponds to a software tool (an IRC client for example). user is an actor in the activity, as he plays a role in it, role that allows him to do actions. An activity can be linked to other ones, when the role of one of its actors implies that this user plays another role in another activity.

The fact that a user plays a particular role in an activity also has an impact on the configuration of his tools. Finally, an activity is an instance of a task, which constitutes an activity model, or pattern. As we said before – and as we will illustrate later – the task is intended to crystallize the experience developed by the actors.

3.2 Choosing Eclipse

Our work in the SD field led us to look closer at the Eclipse platform, which has been adopted by many developers. The success of Eclipse has been a driving force for the development of many tools that can be integrated into the platform as plug-ins. As we have underlined before, we do not aim at developing new tools, but at giving the means to articulate them in a global cooperative activity. From this viewpoint, Eclipse is very interesting since many integrable plug-ins exist.

Basically, the platform is constructed around a core of services called *run-time* that supports the dynamic discovery, installation and activation of plug-ins. A *plug-in* is a component that provides a certain kind of service and respects Eclipse's plug-in specification. It may extend existing features (provided by other plug-ins), implements its own features, and provide extension points (in order to be eventually extended by other plug-ins). This framework allows plug-ins to integrate finely with the environment and other plug-ins. Apart from the minimal kernel managing the basic integration mechanisms, all the proposed functionalities, including the Graphical User Interface (GUI), the workbench, etc., consists of plug-ins or sets of plug-ins that can be managed separately. Thus, from our viewpoint, what makes the success of Eclipse is that it has been principally conceived in terms of tailorability. The end-user can adapt the environment according to his emergent needs. He can discover and dynamically integrate tools that can help him to realize his activity.

Another element in Eclipse, which is in tune with our work inspired by the AT, is the perspectives mechanism. A perspective corresponds to a particular visual point of view on the working environment (and the activated plug-ins) during the realization of a kind of activity. This perspective manages the plug-ins activation and arrangement at the user interface level. Eclipse lets the user create and modify his own perspectives, thus saving his preferences for a kind of activity. From our viewpoint, the perspectives mechanisms provide a powerful mean to crystallize the user's experience. However, one can notice that Eclipse's perspectives can only activate plug-ins that are available on the user's station. In other words, if a plug-in is referenced by a perspective but is not installed, it will be skipped. Another point to notice is that these perspectives are not intended to be shared by users. Even if some people may work with the same perspective because it has been packaged within a specific plug-in, no mechanism has been set up in the environment to give these users the means to share their perspective.

Finally, thanks to its introspection mechanisms, and as we will present it thereafter, Eclipse framework provides very useful means to specify and to support the inter-activities. These mechanisms let us dynamically create the links, until now not supported, that exist between the (sub-)activities supported by the plug-ins in a global cooperative activity. Finally, one must keep in mind that at the time we are writing, as mentioned before, Eclipse does not provide the cooperative dimension we need.

3.3 Managing Inter-activities

Our contribution lies within several levels: first, it consists in extending the Eclipse framework by integrating the elements of our model of activity. According to this model, each plug-in supports one activity. The architecture of the solution we propose is

presented in Fig. 5. We propose to manage the inter-activities thanks to a meta plug-in named CooLDev, whose role is to articulate the other plug-ins in the context of global cooperative activities. This meta plug-in is connected to a CooLDA server, that manages the persistence of the instances of our model. However, each other plug-in – if distributed – is free to use its own communication protocol, server(s), etc. independently.

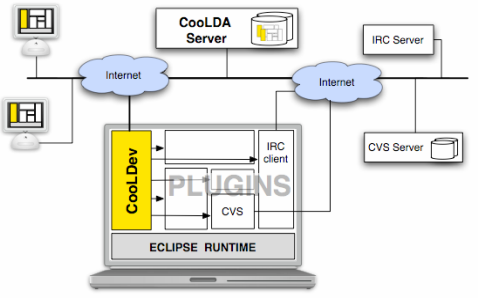


Fig. 5. CooLDev's architecture

The Figure 6 presents an example of such a cooperative global environment for SD that integrates the mechanisms we will describe now.

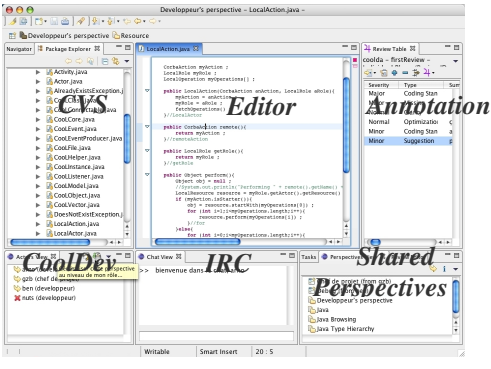


Fig. 6. CooLDev from a particular actor viewpoint

Because of our meta plug-in, the user has first to identify himself to launch Eclipse, in the same way as classical CSCW tools. Thanks to this, we can retrieve information from our CooLDA server concerning the role of the user in the appropriate global activity. This role is an instance of a type of role that can be shared by several actors. It allows then the meta plug-in to configure the user's working environment. To do this, we have extended Eclipse perspectives, in the context of a cooperative activities management: users that play the same role in a particular activity retrieve an instance of the same CooLDev's perspectives. In other words, users playing the same role can get the same environment configuration thanks to CooLDev's perspectives. However, once they have retrieved their perspective, users can adapt it during the activity. As

we will see later, these extended perspectives can also reflect the user's preferences in his role. Moreover, as we underlined in previous part, when an Eclipse perspective was trying to activate an unavailable plug-in, this latter was skipped. In case of a standard use of Eclipse, this is acceptable, since things go otherwise: installed plug-ins are packaged with some perspectives, suitable for their use. In our context, it is the role of the user, and consequently his perspective, which determines which plug-ins will be used in a particular activity. Thus, we have extended the basic perspective mechanism so that it can automatically download plug-ins that are specified in the perspective but not installed on the user's station.

Even if the tools that support the user's activities are instantiated by these extended perspectives, it is not enough to support the inter-activities as we define it. We use our model to specify the actions that have to be processed by a role when its user joins the activity. These actions configure the plug-ins for this user. For example, when a user joins a code reviewing activity, in a development project, CooLDev plug-in uses the user's role to instantiate a CVS plug-in, an IRC plug-in, a code editor and an annotation tool. For the user to avoid identifying another time in the IRC (he has already identified himself while connecting to our environment), we have to indicate to this plug-in what is its configuration (pseudo, server, etc.). From our model's point of view, the user's role realizes actions that trigger operations on the linked plug-in. Technically speaking, the operations are mapped on methods provided by the plug-in and discovered by CooLDA, using introspection mechanisms. We will move later in this paper on the way these links can be set up by users.

3.4 Managing Tailorability and Experience Crystallization

As CooLDev focuses on the inter-activities management, the tailorability it provides sits at this level. First, we take benefit from the tailorability in Eclipse, and we extend it in a cooperative context. Thanks to the plug-ins and perspectives mechanisms, each user can customize the AS by adding, removing and arranging tools that serve his activity. However, in the global cooperative activity, a particular perspective reflects not only the user's preferences, but also his role. Thus, it crystallizes the experience developed in his function. We have set up a mechanism that allows the user to generalize this perspective at the task level, i.e. in his role model. A particular view (Fig. 7) we have developed shows the many actors implied in the activity, their role in it, and whether they are online or not. Behind that, this view proposes the mechanism we have described above: the crystallization of the current user's perspective in his role model. Technically speaking, the CooLDev perspective is sent to the CooLDA server that modifies the appropriate activity model also called task by associating this perspective to the user's role instance. Of course, in the framework of the Coevolution principle (that is a cooperative management of the system adaptation), this action can be proposed only to users with a specific role. One can also imagine that the decision to generalize a particular extended perspective in a role has to be negotiated between actors. Thus, when a new user joins an activity with a particular role, he retrieves the experience of users that have already played the same role, experience that has been gradually constructed and crystallized by these users. Finally, if the scenario of the activity allows it, the user can again modify his own perspective.

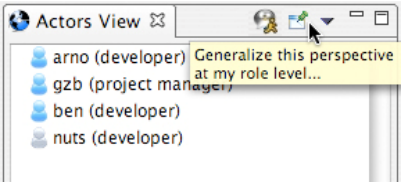


Fig. 7. Zoom on the 'Actors view' and the mean to generalize a perspective at the role level

In order to complete the mechanism that generalizes a perspective in the role model, we have developed a tool (Fig. 8) allowing actors to share their perspectives. This view shows the perspectives shared with others, and allows the users to 'try' the perspectives they receive and to send their perspectives to others. Thus, actors can share extended perspectives without having to crystallize them in their role model, which would be constraining for actors who just want to test perspectives, or to share a viewpoint before deciding to crystallize it. Indeed, the experience crystallization in the roles is an important mechanism, since it may have an impact on all the other actors playing roles that are based on the same model.

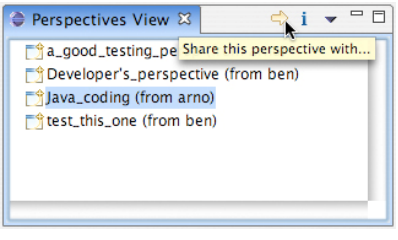


Fig. 8. Zoom on the shared perspectives view

We have also introduced tools that provide tailorability at finer levels of the inter-activities management and help creating links between the tools in our activity model. Thanks to the introspection mechanism, it is possible to (re)define the task during the activity, i.e. the elements that participate in it (for example the actions for each role). The lowest abstraction level is the one of operations that define the links between the actions (of a role) and the methods (provided by a plug-in). CooLDA is able to dynamically retrieve these methods, allowing to (re)define the operations used in the specification of the roles' actions. We are aware that the abstraction level of this kind of tailorability is still hardly within the reach of every end user. However, implementing these mechanisms helps us, at first, to verify the technical feasibility of such an approach. We are now working on user interfaces of higher abstraction level, to give end users the means to access this really fine-grained inter-activities management.

More generally speaking, the evolution of an activity – its tools (integrated plug-ins, operations) and roles (extended perspectives, actions) – is synthesized in the model of activity presented that constitutes the *task*. A task forms thus a model that has crystallized the experience developed cooperatively by users during their global activity, and that can be re-instantiated in order to support other similar activities. These latter, evolving in turn, will be able to modify their task, or to create new ones.

This reflexive approach – even if promising – raises also many problems. We cannot develop them here, but one can refer to [4] to discover the many stakes in it that we have already identified.

The mechanisms described here present the tailorability and crystallization currently provided in our AS. These mechanisms are not at the same abstraction level. As underlined by [17], the tailorability level increases proportionately with the difficulty for end-users to access it. The basic integration of tools and the mechanisms tied to the extended perspectives are more directly aimed at end users. The evolution of roles and actions are aimed at CooLDev specialists (understanding the AS model), and the definition of operations by introspection on methods at developers (understanding object-oriented concept). At first, we will package CooLDev with a set of predefined tasks. Users will then be able to adapt these tasks according to their needs and to their abstraction level, providing thus new activity patterns. The many abstraction levels, and the cooperative dimension of the supported activities let us hope that the end users, interested in the tailorability levels they will be able to reach, will increase their experience towards the system, and will become more and more expert of it, being able to access more advanced levels of tailorability. They will be able then to share this experience through the system, meeting the Coevolution principle.

Of course, this work still needs improvements. Especially, we work on a proposition to extend the plug-ins model, in order to add a semantic level that should facilitate a finer integration by end users. And in order to validate our approach, we plan to experiment the platform in real projects that will help us to verify the stability of the system with regards to its reflexive properties, from a technical and especially a human viewpoint. Indeed, for the system to be 'correctly' adapted, the activity must pass by stable stages, allowing users to increase their experience. So we want to verify that the human dimension brought to the fore in CooLDev (the system is adaptable, but not self-adaptive) permits, not a continuous evolution of the environment, but rather the crystallization of a true experience for end users.

4 Related Work

In the same efforts towards providing tailorable systems by a component-based approach, different works and orientations have been pursued. Even if it would not be possible to present them all, we summarize here those that are similar to our approach.

Some approaches, like those of [12] or [21], propose composition mechanisms that are based on new and original component-based models. Even if interesting, our approach differs in several aspects. Especially, components that can be integrated in such approaches must be developed according to their specifications, which can be constraining and even restricting until the model to be largely adopted. The main difference with us is that we based our application on the Eclipse framework that provides powerful integration mechanisms, and that is widely used. We find on the Internet a great amount of plug-ins that can be integrated in the environment. This critical mass constitutes an interesting repository of tools that can be integrated in our AS in order to support many different cooperative activities.

Teege [23] proposes another approach to achieve tailorability through a variant of components composition called 'feature combination'. A feature is a basic unit that

provides a service. Features are so basic units that in order to integrate a feature, it is sufficient to specify its presence: the idea is to facilitate the integration process so that it can be achieved thanks to simple drag&drop mechanisms for example. Our approach differs in the level of components we integrate. We aim at integrating in CooLDev tools that provide more than basic functionalities but that can support whole activities. Our approach differs also in the levels of integration we aim at proposing. As we explained, we want to provide several tailorability levels, giving means to the users to adapt their environment depending on their skills concerning the system. Thus, we benefit from the integration mechanisms and interfaces of Eclipse – that supports the automatic and dynamic download, installation and activation of plug-ins – that allow a user to simply integrate new plug-ins. Beside this, we have set up more advanced mechanisms and interfaces aimed at more advanced users that help them to (re)define the roles and actions of the AS.

The idea of creating a tailorable CSCW environment while taking foundations in Social and Human sciences has already been developed in [9] with Prospero and in [16] with Orbit. Prospero takes its foundations in Ethnometodology [10] and is a toolkit helping the creation of groupware tools. Orbit is a global and integrated reflective environment taking its roots in the social worlds theory. A strong difference between this work and ours lies in their different theoretical foundations leading to different results. Prospero differs from CooLDA because it offers tailorability to groupware developers, whereas we want – in the long run – to bring it to end-users. Orbit offers reflectivity to end-users through the *Locales* framework but is not interested in dynamic integration of external tools components.

Finally, our approach differs significantly from all the others examined here because it wants to support the crystallization – in our activity model – and re-use of the experience developed by the users.

5 Conclusion

As software development (SD) is today strongly cooperative, we focused in this paper on the means that can support it. We have been working for years in the CSCW research domain, trying to take benefit from SHS theories, especially the Activity Theory (AT). This work has led us to identify the crucial need for tailorability in cooperative environments, and to define the Coevolution principle. By studying several platforms broadly used by developers, we have identified their shortcomings, in line with the stakes defined in the CSCW field. Therefore, we have proposed a solution that consists in an extension of the Eclipse platform which is already broadly used for SD, but which does not integrate the cooperative dimension of such activities at a global level.

Basing on results coming from the AT and on Eclipse properties, we propose a model of activity and a meta plug-in that contextualizes the activities supported by plug-ins. We aim at creating a tailorable support for managing the inter-activities and setting up the Coevolution: the system must support SD cooperative activities and its own cooperative (re)design (meta)activity, while fostering crystallization and sharing of the experience developed by its users. Our proposition brings several levels of

tailorability intended both to end-users and to users with more advanced skills concerning our platform.

Although it already provides a tailorable support for the inter-activities management, our proposition needs to be developed further and to be tested and validated by experiments in real situations. We have to work on raising its abstraction level. In order to achieve this, we plan to pursue our efforts and to look closer at the problem of the semantic associated to components available on the Internet. Indeed, even if solutions trying to palliate this problem exist, one must agree that most of the existing component models are intended to software developers, whereas the results of studies in many fields show that the means for discovering, and dynamically and finely integrating tools would be useful for end users, as it would take into account *in situ* their emergent needs.

References

1. Barthelmess, P., Anderson, K.M.: A view of software development environments based on activity theory. Journal of CSCW 11(1-2), 13–37 (2002)
2. Bedny, G., Meister, D.: The Russian theory of activity, Current Applications to Design and Learning. Lawrence Erlbaum Associates, Publishers, Mahwah, NJ (1997)
3. Booch, G., Brown, A.: Collaborative development environments. Advances in Computers 59 (2003)
4. Bourguin, G.: Les leçons d'une expérience dans la réalisation d'un collecticiel réflexif. In: Actes de la 15éme conférence francophone IHM 2003, pp. 40–47 (2001)
5. Bourguin, G., Derycke, A., Tarby, J.C.: Beyond the Interface: Co-evolution Inside Interactive Systems - A proposal Founded on Activity Theory. In: Proc. of HCI 2001, People and Computer - Interaction without Frontiers, vol. 15, pp. 297–310. Springer, London (2001)
6. Bourguin, G., Lewandowski, A.: Inter-activities management for supporting cooperative software development. In: ISD 2005. Proc. of the 14th Int. Conf. on Information Systems Development, Karlstad, Sweden (2005)
7. Cheng, L., Hupfer, S., Ross, S., Patterson, J.: Jazzing up Eclipse with collaborative tools. In: Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange, Anaheim, California, pp. 45–49 (2003)
8. Christiansen, E.: Tamed by a Rose: Computers as tools in human activity, in [18], pp. 174–198 (1996)
9. Dourish, P.: Using metalevel techniques in a flexible toolkit for CSCW applications. ACM Transaction on Computer-Human Interaction 5(2), 109–155 (1998)
10. Dourish, P., Button, G.: On "Technomethodology": foundational relationships between ethnomethodology and system design. In: Human-Computer Interaction, vol. 13, pp. 395–432. Lawrence Erlbaum Associates, Mahwah (1998)
11. Engeström Y.: Learning by expanding. Orienta-konsultit, Helsinki (1987)
12. Grundy, J., Hosking, J.: Developing Adaptable User Interfaces for Component-Based Systems. In: Proc. of the First Australasian User Interface, pp. 17–25 (2000)
13. Kraut, R.E., Streeter, L.A.: Coordination in software development. Communications of the ACM 38(3), 69–81 (1995)
14. Kuutti, K.: Notes on systems supporting "Organisational context" - An activity theory viewpoint, COMIC European project, D1.1, pp. 101–117 (1993)

15. Lethbridge, T., Singer, J.: Studies of the Work Practices of Software Engineers. In: Advances in Software Engineering: Comprehension, Evaluation, and Evolution, pp. 53–76. Springer, Heidelberg (2002)
16. Mansfield, T., Kaplan, S., Fitzpatrick, G., Phelps, T., Fitzpatrick, M., Taylor, R., Segall, B., Herring, C., Johnson, P., Berry, A.: Toward Locales: supporting collaboration with Orbit. In: Proceedings of Group 1997, Phoenix, AZ (November 1997)
17. Morch, A.: Method and Tools for Tailoring of Object-oriented Applications: An Evolving Artifacts Approach, part 1, Dr. Scient. Thesis Research Report 241, University of OSLO, Department of Informatics (1997)
18. Nardi, B.: Context and consciousness: activity theory and human-computer interaction. MIT Press, Cambridge (1996)
19. Pavlicek, R.G.: Embracing insanity: open source software development. Sams Publishing, Indianapolis (2000)
20. Sarma, A.: A survey of collaborative tools in software development, Institute for Software Research Technical Report, #UCI-ISR-05-3 (2005)
21. Stiemerling, O., Cremers, A.B.: The EVOLVE Project: Component-Based Tailorability for CSCW Applications. AI & Society 14, 120–141 (2000)
22. Suchman, L.: Plans and Situated Actions. Cambridge University Press, Cambridge, UK (1987)
23. Teege, G.: Feature combination: a new approach to tailorable groupware. In: Workshop on Tailorable Groupware, ACM GROUP 1997 (1997)
24. Webster, M.: An end-user view of the collaborative software development market. Market Research Report, IDC #30608, vol. 1 (2003), http://www.collab.net

Managing Service-Based EAI Architectures Evolution Using a Formal Architecture-Centric Approach

Frédéric Pourraz and Hervé Verjus

University of Savoie - Polytech'Savoie
LISTIC-LS - Language and Software Evolution Group
BP 80439, 74344 Annecy-le-Vieux Cedex, France

Abstract. The development of large software applications (like EAI solution) is oriented toward the interoperation of existing software components (like COTS and legacy systems). COTS-based systems are built in ad-hoc manner and it is not possible to reason on them no more it is possible to demonstrate if such systems satisfy properties and/or quality attributes. On the other hand, software architecture domain aims at providing formal languages for the description of software systems allowing checking properties (formal analysis) and to reason about software architecture models. The paper proposes an approach that consists in formalizing, deploying and evolving Service-based EAI architectures. For that purpose, the ArchWare environment and engineering languages (especially the ArchWare formal ADL, based on the π-calculus) and accompanied tools are used.

1 Introduction

Information systems are now based on aggregation of existing and heterogeneous components that have to cooperate in a precise manner in order to satisfy user needs and software functionalities. A new technology has emerged consisting in assembling widely distributed services for building a service-based application, by the way of web standards such as SOAP, XML, WSDL, etc. A set of interacting (Web) services is known as a Services-Oriented Architecture (SOA). One of the main features of SOA is that services involved are autonomous (we will discuss such features latter in this paper), loosely coupled and are widely distributed across the Internet.

Information systems are more and more complex, need more and more functionality provided by several software applications that already exist (COTS or legacy systems). Reusing and assembling existing components (COTS or/and legacy systems) are questions that cope with some difficulties that are not covered by classical component-based programming solutions like EJB, COM+, CCM, etc. As these are specifications for components development, they do not address the case of COTS-based systems, where source code is not available or/and has been previously developed with other specifications and programming languages. The EAI (Enterprise Application Integration) domain provides integration models and techniques for assembling heterogeneous software applications in a pragmatic way. EAI emerging solutions encompass (1) a distributed architecture using web services and (2) a description of the web services centric architecture, expressed using a web services orchestration/choreography language

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 269–280, 2008.

(i.e. XLANG, WSFL, BPEL4WS, etc. [1]). Information systems based on such technology integrate heterogeneous software components, COTS, using a process-based integration approach, where the process description has to insure the execution correctness of the system. Such information systems, building from COTS, will be called COTS-based systems in the following. In such context, an issue is still open: the adequation between the information system provided (i.e. its composition) and the functionalities it would able to provide (i.e. to the end user) and the orchestration of such functionalities according to business processes. Because EAI solutions fail in insuring that the information systems provided succeeds in end-user needs satisfaction, we need new approaches. This paper presents our work in formally describe an EAI solution building from COTS (or legacy systems). The approach used is based on an architecture-centric development process where the system description is the heart of the process. Using such approach, the (abstract) description can be checked, refined in order to obtain more concrete executable descriptions. The paper will also show how architectural evolution is supported. We assume in this paper that COTS are able to interoperate using web services that wrapp COTS' functionalities.

This paper that is a revised version of [2] will first introduce (section 2) a business case coming from a European project. In section 3, we will introduce the formal approach we follow for defining, executing and evolving EAI architectures. Then, section 4 will present the formalization of such architectures, especially from the evolution point of view. Deployment and execution of the generated services oriented architecture will be discussed in section 5, while section 6 will conclude.

2 Business Case: EAI Architecture for Agile Enterprises

2.1 Business Case Scenario

Our business case relates to a company that manufactures a specific product (an axis a mechanical piece). Moreover, this company can subcontract a part of its manufacturing to a subcontractor in order to carry out a specific task.

As you know, enterprises (especially Small and Medium Enterprises) have to be very adaptive in order to satisfy market changes, business changes, customers needs, etc. Agility stands for an enterprise being able to quickly change, adapt their processes according to market changes, without process, end-user and customer service interruption. In such way, the EAI architecture has be able to change dynamically taking account new requirements and new business contracts. Agility interests are manifold and covers some topics among them [3]:

- Increase of compliance with ISO 9001 V 2000 within and between SMEs due to ability to identify business processes;
- Competitiveness increases due to ability to control and adapt business processes;
- Productivity increases due to ability to synchronize business processes (e.g. in just-on-time delivery in the automotive industry) and, as a consequence, increase quality of relationship between SMEs;
- Increase of faith in information system results due to ability to proof business processes properties;

– Investment costs reduction due to ability to integrate existing applications (legacy software or specific applications).

We adopt an illustrating evolution scenario in EAI context. In the scenario, several companies COTS (like ERP, production follow-up software and SPC) are involved thanks to a choreographer, which will orchestrate them according to the wished business process (see Figure 1 - left part):

– An ERP (called *Infodev*) will first send a manufacturing order to the choreographer, which will deploy the different operations in production.
– The operations list will be send to a production follow-up software (called *Alpha3i*) and machines to controlled to a SPC software (called *Arve*).
– The scrap number per operation will be returned to the choreographer, which will next pass this number to the ERP.
– In parallel, a quality control will be performed by the SPC software. We can underline that this operation dont send any information to the choreographer, its a "vertical" application.

Following a planning step on the Contractor level, an overload is detected; this overload is transmitted to the choreographer which starts a process of invitation to tender with various subcontractors. In this case, the previous scenario needs to evolve by adding new business process functionalities in the choreographer (see Figure 1 - right part):

– Within this evolution, the choreographer will receive an overload (higher than 50 pieces in our case) manufacturing order from the ERP (named *Infodev*).
– Then the choreographer will execute in parallel the same scenario as previously described for an order of 50 pieces and a subcontracting process, which implies a new functionality (a negotiation service) to the architecture. This new service consists in finding the best supplier in order to manufacture the production overload.

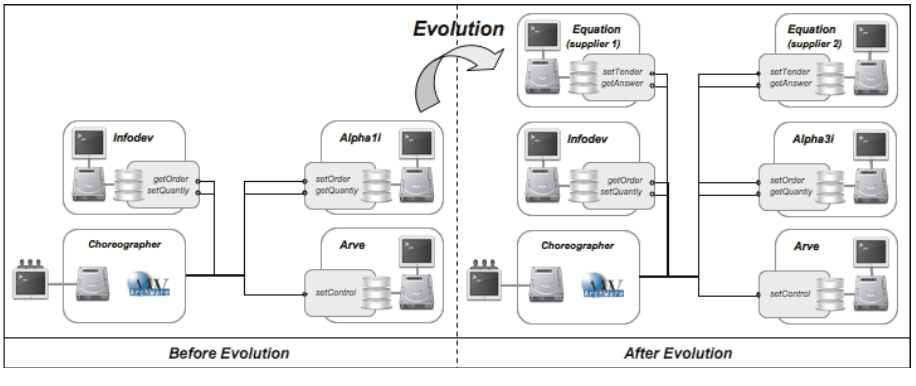


Fig. 1. EAI COTS architectures

2.2 EAI Engineeering Issues

To meet such requirements according to the evolution scenario, best-suited engineering approaches have to be chosen. In the case of complex COTS-based systems (like enterprise information systems in our case EAI) usual and classical engineering approaches fail:

- SMEs need systems that are adapted to their requirements: the design (including properties) of such systems is a crucial step but systems designs/models have to be validated before implemented [3].
- COTS are specific software components with which components classical integration patterns or idioms are not relevant [4] and [5]: COTS have to be characterized as well as their integration. COTS-based system models (when existing) cannot be checked nor validated. That is, one cannot reason on models nor analysis can be made on such models. This lack of formalization has following consequences:
- Design (of complex systems) expertise cannot be caught nor maintained;
- There is a gap and discrepancies between the design and the execution. It is impossible to guarantee that the execution will be conformant to the design;
- The COTS-based systems evolution (substitution, deletion, addition of COTS, changing system behaviour, process, etc.) is not well supported nor it can be validated;
- Crucial properties (safety, completeness, consistency, etc.) of the systems are not taken into account.

As EAI solutions are located in a distributed context (distributed enterprises, networked enterprises,), the EAI architects are enthusiastic by using web services as technology for supporting COTS interoperability. SOAs have to deal with many of the issues encountered in more classical COTS-based systems. Web services will be employed as COTS facets (also called wrappers) that will be orchestrated in order to satisfy EAI goals (the production a mechanical piece: an axis in our case). Web services are accompanied by standards that support part of the interoperability (i.e. SOAP protocol, WSDL, etc.). We propose to follow an architecture-centric approach taking into account evolution and proposing some solutions that might meet the needs issued by the previously identified limitations.

3 A Software Architecture-Centric Approach for Fomalizing EAI SOA

3.1 Architecture-Centric Approach

The architecture of software intensive system (such as an EAI architecture) defines the elements that compose the system, and how they interact [6,7]. The software architecture definition can be made informally, or by using a dedicated language. Different abstraction levels are considered for describing the software architecture [8]. The use of formal architecture refinement guarantees the preservation of properties specified at abstract levels all the way towards architecture implementation.

The architecture centric development process (see [2,9] and [10] for more explanations and that process) aims at providing means for defining software intensive systems at a very abstract level. Such descriptions can be then validated in order to check systems properties and are refined in a more concrete description (that allows to deploy the system in a concrete environment).

The architecture-centric development process we propose is quite different to the classical software development process (i.e. waterfall, spiral, etc.): if the system behaviour does not fit the requirements, the architecture description can be modified without restarting entirely the development process. Representations (architectural descriptions) are also checked at every stage of the process before generating the code.

The work on architecture centric approaches for software development has been very fruitful during the past years, leading, among other results, to the proposition of a variety of Architecture Description Languages (ADLs), usually accompanied by analysis tools. The enthusiasm around the development of formal languages for architecture description comes from the fact that such formalisms are suitable for automated handling. These languages are used to formalize the architecture description as well as its refinement. The benefits of using such an approach are manifold. They rank from the increment of architecture comprehension among the persons involved in a project (due to the use of an unambiguous language), to the reuse at the design phase (design elements are reused) and to the property description and analysis (properties of the future system can be specified and the architecture analyzed for validation purpose).

Different ADLs have been proposed [11]. According to our requirements we need a formal ADL that will support: the description of the architectural structure and behaviour, the properties expression and dynamic evolution [9]. We also need a way to express processes. We will introduce the ArchWare approach that combines interesting formal features in a unified ADL.

3.2 The ArchWare Environment

The main objective of the ArchWare project (European IST-5 project, number 32360) was to provide the necessary elements to the engineering of evolvable software systems. In order to achieve this goal, the ArchWare project developed an integrated set of languages and architecture centred tools while being based on a persisting execution framework [10]. The ArchWare project provides some engineering technologies, among them:

- Innovating languages centred architecture (architecture description language and analysis language),
- Refinement models,
- Customizable software environments and tools dedicated to the engineering of evolutionary software systems.

ArchWare aimed at building a customizable environment of engineering software, which can be used to create software architecture centred environments (Oquendo et al., 2004). This project considered that a customizable architecture centred environment is structured in two distinct layers, namely a runtime framework and a set of architecture centred tools. The ArchWare runtime framework [10] includes an execution engine of

architectures based on evolutionary processes of development, a refinement process of architecture description and mechanisms supporting the interoperability of the environment tools and components (that can be COTS). Details of the ArchWare environment can be found in [12]. The ArchWare architecture centred tools provides supports for:

- The definition of the architecture,
- Validation of such architectures (using analysis tools and software graphical animation tool),
- The checking of the functional and extra functional properties of architectures,
- The refinement of architecture descriptions from an abstract level to a concrete level,
- The code generation of the systems in various programming languages (using explicit rules).

3.3 Architecture Evolution Support

One of the ArchWare environment key features is the evolution support ability [13]. On one hand, ArchWare ADL is the language allowing to describe evolvable architectures (i.e. architectures that can dynamically evolve); on the other hand, the ArchWare environment contains an ADL virtual machine [10] that supports dynamic evolution (the architecture description code can be modified while being interpreted). Then, an architecture description can be dynamically changed and the runtime architecture change accordingly (we will present an evolution scenario latter in this paper). When an architecture evolves dynamically, one may check the new architecture against properties or not (it is up to the architect).

According to our needs, the ArchWare ADL [14] is the only one formal ADL (section 2) that:

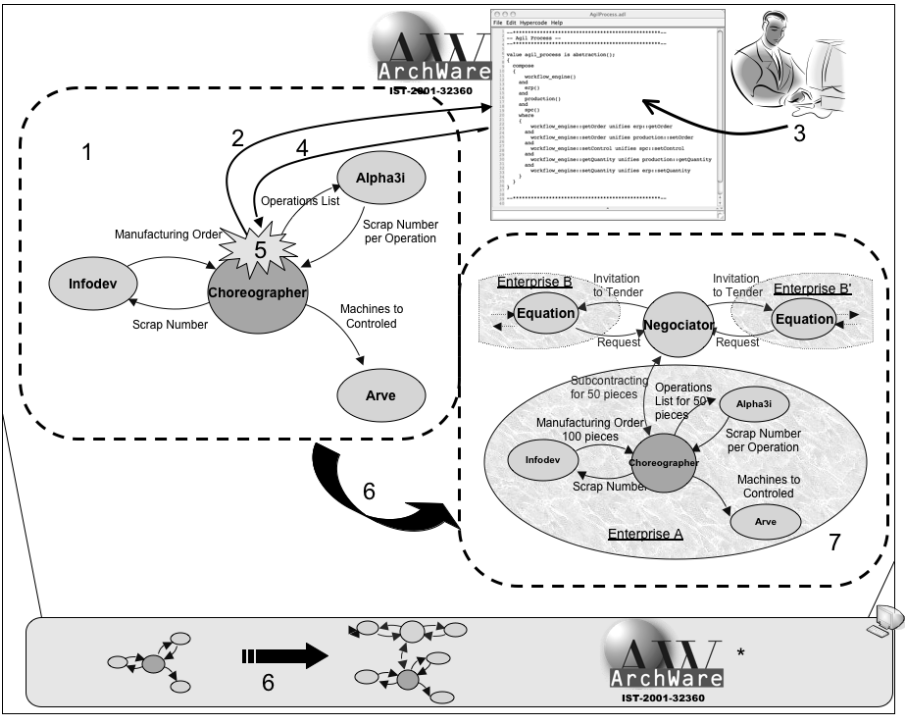
- Allows the architecture structural modelling as well as the behavioural description (as an extension of π-calculus [15]);
- Supports properties/constraints definition;
- Supports dynamic evolution of the architecture.

We will also propose a way to enact services oriented architecture as the concrete EAI architecture, following the refinement approach.

4 EAI Architecture Evolution Formalization

We decided to describe EAI architectures using ArchWare ADL. According to the architecture-centric process we adopt, the formal descriptions can then be refined in order to obtain a concrete representation (Figure 1). At the early stage of the development process, abstract architectures have to be expressed according to domain specific characteristics and ilities.

The EAI architecture description contains the code which represents the first stage of the architecture (Figure 2 - left part), with the code allowing the architecture to request an evolution in order to behave as the architecture shown in Figure 2 (right part). The



1. *The architecture behaves conform to the description*
2. *During execution, when some (unpredictable) situation occurs, the choreographer requests the ArchWare editor for an evolution*
3. *Using the editor, the architect expresses the evolved architecture…*
4. *…that is sent to the choreographer*
5. *The choreographer is now applying the provided behaviour that is dynamically interpreted by the virtual machine…*
6. *…and the architecture is changing dynamically…*
7. *… and is now behaving conform to the provided behaviour*

**. All of this is supported by the ArchWare environment*

Fig. 2. EAI architecture evolution

evolution has also to be expressed at the architectural level (the evolution is described using the ArchWare ADL code it is part of the EAI architectural description) in order to deal with the evolution that occurs at the enterprise level.

Main pieces of EAI architecture description code (in ArchWare ADL) including evolution expression are presented hereafter. The code contains both the classical architectural description in terms of elements (often called processes or components in ADL) interacting each other, some properties according to EAI domain and business process code.

It is innovative as the ArchWare ADL language allows to formalize several facets of an EAI architecture (topology, properties, business process and evolution that can focus on part or all of the architectural artefacts).

The ERP COTS (Infodev) can be defined in ArchWare-ADL as follow:

```
value erp is abstraction(); {
  value getOrder is free connection(String,
  String, Integer, String);
  via getOrder send "order-1", "axe", 100,
```

```
  "JUN 17 2005";
  value setQuantity is free connection
  (String, String, String, Integer);
  via setQuantity receive store:String,
  code:String, article:String,
  quantity:Integer;
  done }
```

According to ArchWare ADL concepts [14], each COTS is described as an abstraction which declares some connections on where parameters can be sent and/or received.

The ERP COTS has got two sequential actions. First, it sends the description of the new order via the *getOrder* connection. Secondly, it will receive the production report via the *setQuantity* connection. According to the ArchWare ADL syntax, the keyword *done* stands for the terminate action (Tau in π-calculus). As well as the ERP COTS, both production report and SPC COTS are designed by using abstractions. As previously described, they define some connections allowing them to send and to receive parameters.

The definition of the choreographer is also based on an abstraction description but contains more complicated actions. After receiving an order via the *getOrder* connection, the choreographer composes two processes in parallel:

1. The first one is the internal manufacturing process. It first calculates the right quantity to product (100 or the initial quantity if it is less than 100). After that, it sends the order to product and to control (via *setOrder* and *setControl*), receives the production report and transmits it to the ERP (via *getQuantity* and *setQuantity*);
2. The second one requests an evolution in case of overload detection. It first tests the quantity and if an overload is found, it will send a request to a particular ArchWare tool: Hypercode-Editor [10] (via *hypercode_request*). The end-user architect is asked to provide a new abstraction that corresponds (its behaviour) to the subcontracting process (see section 5). This latter is received (remember that one of the powerful features of the π-calculus is that processes can be exchanged between other processes) and instantiated by the choreographer (via *hypercode_reply*).

```
value choreographer is abstraction(); {
  value getOrder is free connection(String,
  String, Integer, String);
  via getOrder receive code:String,
  article:String, quantity:Integer,
  date:String;

  compose {
    behaviour {
      if(quantity > 50) then {
        value newQuantity is 50; }
      else {
        value newQuantity is quantity; }
      value setOrder is free connection
      (String, Integer, String, String);
      value setControl is free
      connection(String);
      via setOrder send code, quantity,
      article, date;
      via setControl send code;
      value getQuantity is free
      connection(String, String, Integer);
      via getQuantity receive code:String,
```

```
      article:String, quantity:Integer;
      value setQuantity is free connection
      (String, String, String, Integer);
      via setQuantity send "stock", code,
      article, quantity; }
    and behaviour {
      if(quantity > 50) then {
        value hypercode_request is free
        connection();
        via hypercode_request send;
        value hypercode_reply is free
        connection(abstraction(String,
        Integer));
        via hypercode_reply receive
        subcontracting_process:
        abstraction(String, Integer);
        subcontracting_process(article,
        quantity-50); } } } }
```

Finally, the following EAI abstraction that corresponds to the bootstrap of the EAI architecture must be defined. This abstraction instantiates, in a single process, the COTS abstractions previously defined and unifies all of the connections.

```
value eai is abstraction(); {
  compose {
    choreograher() and
    erp() and
    production() and
    spc()
    where {
      choreograher::getOrder
      unifies erp::getOrder
      and ... } } }
```

Once defined, the EAI architectural description can be analyzed (see [2]). Then, the architecture is deployed as a services oriented architecture and can be interpreted by the ArchWare runtime environment. One can note that we are now able to easily adapt such architecture for other EAI solutions (by modifying ArchWare ADL code).

The next section will show the refinement consisting in generating web services WSDL code from a ArchWare ADL specification.

5 Deploying, Executing and Evolving EAI Architecture as a SOA

5.1 Architecture Deployment and Execution

At the final stage of the refinement (the concrete architecture), we obtain a Services Oriented Architecture where web services are used as COTS facets. The web services allow EAI components (COTS and legacy systems) to interoperate (Figure 2). In such concrete context, all well-known languages (WSFL, XLANG, BPEL4WS, etc.) and web-based technologies (WSDL, SOAP, etc.) may be candidates for supporting the deployment and the execution of our systems using web services.In our particular case, the business process is part of the entire architecture and expressed using ArchWare ADL (the choreographer is an abstraction in term of ArchWare ADL and its behaviour is the business process).

As we introduced previously, the ArchWare runtime environment enacts the entire architecture (including business processes). External components (COTS) will interoperate with the ArchWare environment through web services while the choreographer

will be part of the ArchWare environment (enacted by the ArchWare virtual machine). From the architectural abstract description we generate the COTS web services concrete description (the WSDL code is basically obtained from our abstract description of services the COTS provide - their APIs and only if the web services do not exist). Generic refinement rules (presented in [2]) support the transformation from ArchWare ADL to WSDL. All other elements of the abstract architecture are enacted by the virtual machine.

5.2 Architectural Dynamic Evolution

The concrete architecture can now be interpreted. According to the business scenario we presented in section 2, the EAI architecture is behaving as the one shown in Figure 1 (left part). Then, when a production capability threshold is reached, suppliers have to be added in order to satisfy the new production demand. Then, the EAI architecture is now the one presented in the Figure 1 (right part) and is behaving as the latter (see the Figure 2 for the overall evolution process). The evolution concerns:

- The architecture topology (by addition of suppliers several abstractions in ArchWare ADL concepts);
- The communication between architectural elements (i.e. connections between abstractions);
- A new business process with several enterprises and more COTS involved.

Note that the concrete architectures are not symmetric (in term of number of architectural elements) to the formalized architecture (in ArchWare ADL code). This is due to that more architectural elements (abstractions) are necessary to be defined in order to provide functional and non-functional architectural aspects required by EAI architectures. The concrete architectures are only composed by all of the required web services (one per COTS) plus the ArchWare environment that enact the architecture (both the enacted architecture and the ArchWare architecture centred tools).

6 Conclusion and Ongoing Work

Building COTS-based system generally fails due to non-formal approaches (often ad-hoc solutions like in classical EAI engineering approaches) used. In [4] and [16,5] we claimed that designing and building COTS-based systems addresses lots of issues: the gap between the design level and the implementation one and the evolution support are two of them. Because COTS (as well as legacy systems) already exist, we focus to the glue between such software components (COTS, etc.). The approach presented in this paper innovates by providing a formal approach for the development, deployment and enactment of an EAI architecture as well as its dynamic evolution. This approach is an unified approach consisting in refinement steps from specification to implementation code generation. Our approach is divided in two parts: (1) the definition of an architecture that is convenient for the design of COTS-based systems as well as it is also closed to a concrete architecture (in our case, a SOA) and (2) an architecture-centric development process using a formal ADL as a specification language (that deals with structural aspects and behavioural aspects - including business processes).

Actual web services composition languages [1] are semi-formal and does not deal with formal properties or QoS, other approaches, more formal (i.e. [17]) introduce gap between description level and execution level. We currently work on providing a SOA specific formal language and dedicated environment (called Diapason [18]) for building, executing evolvable SOAs and reasoning about them.

References

1. Peltz, C.: Web services orchestration: A review of emerging technologies, tools, and standards
2. Pourraz, F., Verjus, H., Oquendo, F.: An architecture-centric approach for managing the evolution of eai services-oriented architecture. In: ICEIS 2006. Eighth International Conference on Enterprise Information Systems, Paphos, Cyprus, pp. 234–241 (2006)
3. dit Jolicoeur, L.B., Braesch, C., Dindeleux, R., Gaspard, S., Berre, D.L., Leymonerie, F., Montaud, A., Chaudet, C., Haurat, A., Theroude, F.: Final specification of business case 1, scenario and initial requirements. Deliverable D7.1b, ArchWare Consortium (2002)
4. Estublier, J., Verjus, H., Cunin, P.Y.: Designing and building software federations. In: CBSE - EUROMICRO 2001. 1st Conference on Component Based Software Engineering, Warsaw, Poland, pp. 121–129 (2001)
5. Cimpan, S., Verjus, H., Oquendo, F.: Cots-based system design using software architectural styles. In: Integrated Design & Process Technology (Special Topic Session Formal Reasoning on Software Components and Component-based Software Architectures), Austin, USA, pp. 127–134 (2003)
6. Garlan, D., Shaw, M.: An introduction to software architecture. In: Ambriola, V., Tortora, G. (eds.) Advances in Software Engineering and Knowledge Engineering, pp. 1–39. World Scientific Publishing Company, Singapore (1993)
7. Shaw, M., Garlan, D.: Characteristics of higher-level languages for software architecture. Technical Report CMU-CS-94-210, Carnegie Mellon University, School of Computer Science (1994)
8. Allen, R.J.: A Formal Approach to Software Architecture. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh (1997)
9. Verjus, H., Cimpan, S., Alloui, I., Oquendo, F.: Gestion des architectures évolutives dans archware. In: CAL 2006. 1ère Conférence francophone sur les Architectures Logicielles, Nantes, pp. 41–57 (2006)
10. Morrison, R., Balasubramaniam, D., Kirby, N., Mickan, K., Oquendo, F., Cimpan, C., Warboys, B., Snowdon, R., Greenwood, M.: Support for evolving software architectures in the archware adl. In: 4th Working IEEE/IFIP Int. Conf. on Software Architecture, Oslo, Norway, pp. 69–78 (2004)
11. Medvidovic, T.: A classification and comparison framework for software architecture description languages. IEEE Transactions on Software Engineering 26(1), 70–93 (2000)
12. Consortium, A.: Archware - architecting evolvable software - ist european project 2001-32360
13. Cimpan, S., Verjus, H.: Challenges in architecture centred software evolution. In: CHASE: Challenges in Software Evolution, Bern, Switzerland, pp. 1–4 (2005)
14. Oquendo, F., Alloui, I., Cimpan, S., Verjus, H.: The archware adl: Definition of the abstract syntax and formal semantics. Deliverable D1.1b, ArchWare Consortium, ArchWare European RTD Project IST-2001-32360 (2002)
15. Milner, R.: Communicating and Mobile Systems: The π-calculus. Cambridge University Press, Cambridge (1999)

16. Verjus, H., Cimpan, S., Telisson, D.: Formalising cots-based federations using software architectural styles. In: ICSSEA 2002. 15th International Conference on Software & Systems Engineering and their Applications, Paris, France, vol. 2, pp. 1–8 (2002)
17. Salaün, G., Bordeaux, L., Bordeaux, M.S.L., Schaerf, M.: Describing and reasoning on web services using process algebra. In: ICWS, pp. 43–50. IEEE Computer Society, Los Alamitos (2004)
18. Pourraz, F., Verjus, H.: Diapason: an engineering environment for designing, enacting and evolving service-oriented architectures. In: ICSEA 2007. Proceedings of the International Conference on Software Engineering Advances, IEEE Computer Society, France (accepted for publication 2007)

Agile System Analysis and Design*

Yael Dubinsky[1], Orit Hazzan[2], David Talby[3], and Arie Keren[3]

[1] Department of Computer Science, Technion – Israel Institute of Technology
yael@cs.technion.ac.il
[2] Department of Education in Technology & Science,
Technion – Israel Institute of Technology
oritha@tx.technion.ac.il
[3] MAMDAS – Software Development Unit, Air Force, IDF, Israel
davidt@cs.huji.ac.il, ariekk@netvision.net.il

Abstract. Agile software development methods mainly aim at increasing software quality by fostering customer collaboration and performing exhaustive testing. The introduction of Extreme Programming (XP) – the most common agile software development method – into an organization is accompanied with conceptual and organizational changes. These changes range from daily-life changes (e.g., sitting together and maintaining an informative project environment) and continue with changes on the management level (e.g., meeting and listening to the customer during the whole process and the concept of the whole team which means that all role holders are part of the team). This paper examines the process of transition to an agile development process in a large-scale software project in the Israeli Air Force as it is perceived from the system analysis and design perspective. Specifically, the project specifications of the agile team are compared with those of a team who continues working according to the previous heavyweight method during the first half year of transition. Size and complexity measures are used as the basis of the comparison. In addition to the inspection of the specifications, the change in the role of the system analysts, as the system analysts conceive of it, is examined.

1 Introduction

System analysis and design are basic activities in software development. Traditionally, they are carried out by a separate group of practitioners, who gather the system requirements, analyze them and prepare the specifications documents to be handed to the development group. In large-scale software projects these activities are highly significant.

The agile development methods [6] and specifically Extreme Programming (XP) [1, 2] introduce a change in the software development environment. For example, working according to primary practices of XP, the team *Sits Together* while implementing

* This chapter is based on the following paper: Dubinsky Y., Hazzan O., Talby D., and Keren A.: System Analysis and Design in a Large-Scale Software Project: The Case of Transition to Agile Development. In the 8th International Conference on Enterprise Information Systems (ICEIS), Paphos, Cyprus (2006).

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 281–292, 2008.

the notion of *Whole Team*, which means that role holders, such as the system analyst, work with the other role holders – developers, testers and team leaders. These work habits are introduced to foster communication among teammates.

A natural question to be asked at this point is: How can we deal with these notions in a large-scale project? Should system analysts need to *Sit Together* with the development team? What is the *Whole Team* with respect to system analysis and design and how this notion is interpreted in a large-scale software project?

Another example is the XP primary practice of *Weekly Cycle*. According to this practice, the work is planed on a weekly basis in accordance with full customer collaboration. In this case, we should ask: What is the role of the system analysts in these weekly planning sessions? Are system analysts the mediators or do they listen to the customer together with the *Whole Team* (i.e., developers, testers, and so on)? How do we expect the project specifications to be expressed in an agile environment?

This paper presents a field research conducted in a large-scale software project in the Israeli Air Force. The research examined the process of transition from heavyweight software development to agile development. Focusing on the system analysis and design aspect, the research aims at answering questions such as above-mentioned ones.

In Section 2 we elaborate on the transition process and in Section 3 we explain the research setting for its investigation. In Section 4 we present data analysis by comparing the agile project specifications with those of a team which continues working according to the previous heavyweight method. The comparison relates to the first half year of transition. Size and complexity measures are used for the comparison. In addition, in Section 4 we present data and analysis with respect to the change in the role of the system analysts as they conceive of it. In Section 5 we conclude.

2 The Transition Process

The in-transition software project that this paper focuses on has been developed by about eighty skilled system engineers, system analysts, developers and testers, organized in a hierarchical structure of small teams. The project develops large-scale, enterprise-critical software, intended to be used by a large and varied user population.

The army is known as a large and hard-to-change organization with respect to fixed regulations, project approval, management methods and organizational structure and culture. However, when the project leadership decided to change the software development method in order to cope successfully with the challenges that the project set, the Air Force leadership supported the decided-upon transition as a mean to improve software process and quality.

After several months during which the fitness of different development methods to the said project had been investigated, XP was selected to be implemented and a pilot team of fifteen people was established and started working according to the agile method. All the other teams of the project continued working according to the previous heavyweight method.

It is important to note that during the years prior to the transition, tools and procedures were developed and used by the people in this software unit. Though it was accepted that agile development can improve the process, it was also agreed that there are tools and procedures that will not be changed at the current stage, whether because they are good practices or whether because of time constraints.

The software project is built based on a large-scale in-house object-oriented framework [8], which handles many of the underlying technical aspects of the system. One aspect is the *formal detailed specifications*. This framework relies on a metadata repository [12], which contains most of the system's specifications: data entities, data types, actions, transactions, user types and privileges, messages, external interfaces and so forth. This data is edited in the repository, in formal forms – in contrast to free-text documents – and much of it is used to automatically generate code and other files.

As a result of working with this framework, the process of development starts with design, continues with writing the formal detailed specifications in the metadata repository, and then coding those parts of the specifications that are not automatically generated. In such a process, the specification writers have to be formal and precise, and as formality increases, the cost of communication increases when teams later on communicate in order to clarify loose ends.

During the transition process all teams in the project, including the agile team, continue working with formal detailed specifications and with the respective tools that support them.

The roles involved with system analysis and design in this project are architects, operational system analysts, functional system analysts, and system engineers. In this work we focus on the operational and functional system analysts. The operational system analysts are practitioners in the operational aspects of the project subject matter and are part of an operational analysis group. They define the system to be developed and they represent the customers and users. The functional system analysts process the operational specifications and convert them into engineered technical specifications. They are part of the development group.

The change for role holders stems from the change in process. As part of the transition process, only operational and functional meta-specifications are produced, and then delivered to the agile team who together with the customer and system analysts produced the detailed specifications for both operational and functional aspects. The impact of this process on system analysts is elaborated in Section 4.

Before we delve into the details of the research findings, we present the difference approaches reflected by the two development approaches as they were described by one of the system analysts who was involved with the transition process in general and with the agile team in particular. Using the metaphor of trips he says that the heavyweight software method is like an organized tourist trip while the agile method is more a journey-like trip. Specifically, the tourist makes decisions long time before executing them, plans ahead into the small details, and has not much tolerance for changes; The journeyer, in contrast, is flexible and open to changes, makes decisions closer to their carrying out, knows in general terms what he/she wants to see and plans the details only during the journey itself.

3 Research Framework

The exploration of this transition process started two years ago when it was decided to change the traditional heavyweight software method that had been used in this organization for many years. In previous work, we presented the way agile and XP were introduced into this project [3] together with the set of product and process metrics

evolved in the first release of the pilot team and that guided in practice the development method [4].

Within this research, the sub-research that focused on the expression of the system analysis and design aspects in the transition process, two research approaches were used.

The first approach is a quantitative comparative one, by which we aimed at measuring the implications of the transition to the agile method on system analysis and design. Accordingly, we examined and compared the specifications produced from both kinds of teams – the traditional one and the agile one. Thus, one of the main contributions of this research is the comparative data and field-based evidence it provides with respect to the role and functionality of system analysis and design in an agile XP large-scale project in a large organization.

The second research approach was a qualitative approach in which we seek to understand the process from the system analysts and designers' point of view. Accordingly, we interviewed system analysts and asked them questions such as "Do you feel that your role has been changed? If no, please describe your role before and after the transition. If yes, please describe how your role has been changed.", "Please compare the traditional way with the agile XP one.", and so forth.

In what follows the research tools are presented. For the comparison purposes, we look at two different sets of specifications. The first set belongs to the team which worked according to the heavyweight method and during the examined half year was in the phase of fault corrections before delivery. The second set belongs to the team which worked according to the agile method and during the examined half year developed three release – the second, third and fourth releases – which were each two months long, and composed of 4 two-week iterations.

It is clear that a comparison of the specifications of two different products of two different teams is not a trivial matter. Therefore, we searched for trends and relative-to-size measures rather than absolute numbers. In addition, the comparison value increases because the two teams work in the same organization, according to the same procedures, with the same infrastructure and tools, and with people with similar experience and expertise.

Three measures were taken from each set. The first measure is the size of the specifications which is used for comparison alignment. The second and third ones are two measures which are used to assess the complexity of the specifications; one of them is inspired by the measure of code cyclomatic complexity [7, 14]. In Section 4.1 these measures are elaborated and illustrated.

In order to learn about the transition process from the point of view of the system analysts, three thorough interviews were conducted. The interview was composed of five parts as follows. The first part was an introduction in which we explained the goals of the research and the interview, ask permission for videotaping, and answer questions if exist about our research and about the interview. The second part concerned with the interviewee's current position in which we asked to describe the current role and the significant and interesting things as well as the problems that occurred as part of this role before and after the transition. The third part addressed the system analyst role in general in which the interviewee was asked to define the role, draw the position of this role in the organization, and reflect about the drawing. The forth part of the interview focused on the agile environment. The interviewee was asked to share with us his/her knowledge about the agile method in general and its agile

implementations in the organization in particular. Then the role of the system analyst was discussed with respect to the agile environment and with respect to the drawing from the previous part. In the last part of the interview, the interviewee was asked to imagine that she or he are going to establish a new software company and to decide about the desired skills and education of the system analysts who they will hire.

The qualitative data that is the outcome of these interviews was interpreted by using the theory of coping with change [9] and by using a reflection ladder [5, 10, 11, 13].

4 Data Analysis

The data presented and analyzed in this section was gathered as described in Section 3. In Section 4.1 the project specifications are compared using size and complexity measures. In Section 4.2 the change in the role of the system analysts is examined.

4.1 Specifications Comparison

The examined specifications are divided into modules. We denote the specifications of the team which worked according to the heavyweight method by *SpecH* and the specifications of the team which worked according to the agile method by *SpecA*. During the half year we took three main measurements – in the beginning, after two months, and in the end (after 6 months). *SpecH* was composed of 189, 196 and 200 modules in these three measurement times respectively. *SpecA* was composed of 34, 44, and 56 modules respectively.

The specifications are written in formal documents, to enable automatic code generation. Figure 1 shows an example of specification fragment. The first measure we use was size, meant intuitively to represent the number of decisions made in the specifications. Therefore, a size of 1 is given to every simple specified value (such as minimal value and maximal value), and a size of 1 is given to each line of free-text specifications. Therefore, the size of the fragment in Figure 1 is 12 since it has 6 simple values, 1 for the one line of the 'Is Required' specification, and 5 for the five lines of the 'Do on change' specification.

Field Name:	Name
Field Type:	String
Description:	The customer's full name
Minimal Length:	1
Maximal Length:	40
Field Editor:	Text Box
Is Required:	*only* ***if*** *the ID field is empty*
Do on change:	***If*** *the ID field is non-empty,* *check that it matches the new name.* ***If*** *match* *enable the 'OK' button,* ***else*** *display the 'Name/ID Mismatch' error message.*

Fig. 1. Specifications sample

Since the simple values in a specification result in generated code, and hence do not require any coding, the size measure does not reflect the complexity of a given specification for the development team.

Complexity is only created by the free-text specifications – and to represent this, we devised two complexity measures. The first is the *Logic-Based Complexity* that is calculated by counting the number of lines of non-trivial specifications. For the specification shown in Figure 1, this measure would be 6. The second is the *Keyword-Based Complexity* that is inspired by the cyclomatic complexity measure [7, 14], in which a sequential method has a complexity of 1, and each decision that causes a split into two directions raises the complexity by 1. This definition is equivalent to defining the complexity as the number of paths in the method's decision graph. We emulate the cyclomatic complexity measure by defining the complexity of free-text specifications paragraphs to be 1 and add the number of appearances of the following popular keywords: if, else, for-every, for-each. For the specification in Figure 1, this measure would be 6 since we count 2 from the 'Is Required' specification (1+1 occurrences of 'if'), and 4 from the 'Do on change' specification (1+2 'if'+1 'else'). Validating with the specifications, we found this emulation as a good and sensible approximation of the actual number of paths in the specification. Although these specifications are free-text, the analysts writing the specifications normally use only these words. They are often manually marked by making the font bold, as shown in Figure 1. This is a project-wide practice, ensuring the quality of data.

Figure 2 presents the logic-based complexity of *SpecH* and *SpecA*, averaged over all modules of each project. As can be observed, the averaged logic-based complexity in the *SpecH* project is four times higher than that of *SpecA*. This difference is important since we expect that both the agile development will continuously simplify *SpecA*

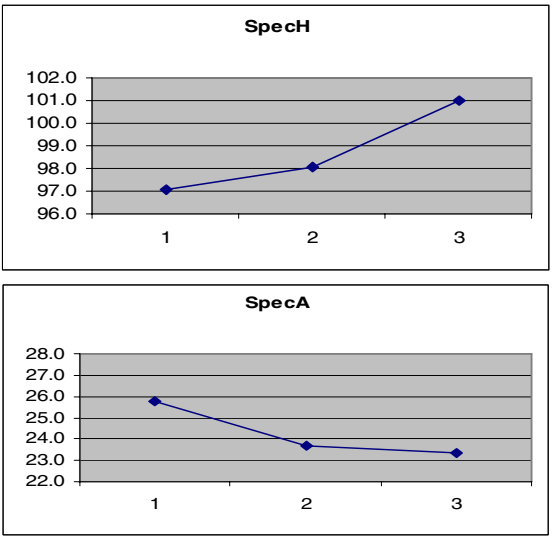


Fig. 2. Averaged logic-based complexity in three check points

(due to continuous refactoring) and that the stabilization phase of the heavyweight development will simplify *SpecH*. In addition, we can see that the average logic-based complexity of SpecH *increased* by 4% during the researched six-month period, while the same metrics *decreased* in SpecA by almost 10%.

Figure 3 presents the averaged keyword-based complexity of *SpecH* and *SpecA*. As can be seen, the logic-based complexity and the keyword-based complexity are highly correlated. Also, as in the previous case, in both cases, the values of the averaged keyword-based complexity per each specification in each measurement point are similar, though for *SpecH* the range of values is 3.5 times higher than that of *SpecA*. The trend of change over time is also similar to that observed for logic-based complexity.

This difference in trends over time can be attributed to the different development methods. Note that this difference is for the average complexity over all modules, so the absolute size of each project is irrelevant here. The heavyweight project is in a mature phase; Although 11 modules were added to it during the researched period, the average per-module complexity increased, hinting that most new functionality was embedded in existing modules.

In the agile project, on the other hand, 22 new modules were added, and average complexity noticeably decreased. According to the team's testimony, and to XP practices, this is caused by continuous refactoring. When a module gets too complex, it is refactored into (possible several) simpler modules. The goal is to keep the design simple over a long period of time, not assuming the "right" design in advance. In contrast, in traditional projects, the design of modules is usually set in advance. It may also be the case that the high absolute complexity of the heavyweight project, achieved over time (as it is in a more mature state than the agile project), makes refactoring at this stage more expensive and risky.

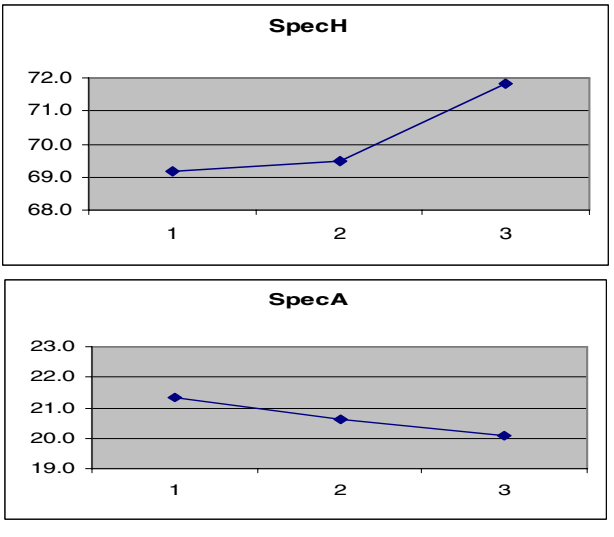


Fig. 3. Averaged keyword-based complexity in three check points

The difference in absolute complexity can be explained by several factors that do not stem from the development method. For example, the agile project could be inherently simpler than the heavyweight one. From conversations with people in both projects, this is definitely not the case, and two other explanations have been proposed. First, that the agile project reuses more features that are built into the framework, and can be specified in a way that enables automatic code generation. And second, the experience gained from specifying the (earlier) heavyweight project was exploited to specify the agile project in a way that enables greater use of the framework, and less manual coding. Figures 4 and 5 support these explanations, by presenting the ratio of logic-based and keyword-based complexity to the size measure, thereby measuring the proportion of complex to simple specification. The values are about 50% and 20% lower in the agile project, for these two measures respectively.

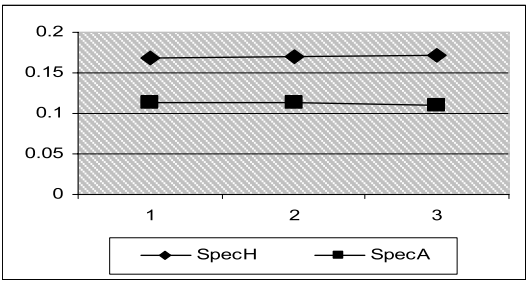


Fig. 4. Logic-based complexity to size in three check points

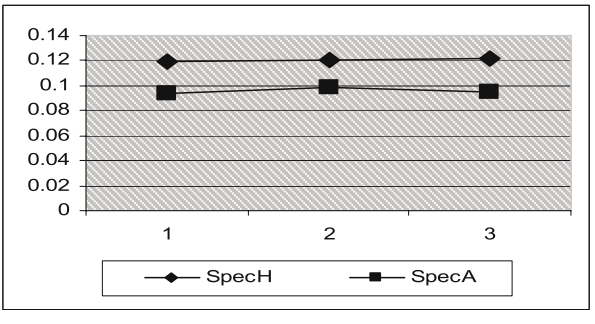


Fig. 5. Keyword-based complexity to size in three check points

We elaborate with the notion of feedback cycles which can add further explanation to the results described. Before starting using the agile practices, system analysts had little opportunities to receive feedback on their work. The specifications were produced weeks before implementation. The analysts continued to develop next functions before they saw the implementation of the previous one. The developers could not

contribute their remarks to the specifications. This is not the situation with the agile team. The work of the system analysts is examined regularly on a two-week basis. The analysts receive continuous feedback on their work and can navigate the design accordingly.

4.2 The Role of System Analysts

Based on the qualitative data gathered in the interviews, we focus in this subsection on the system analyst role and the changes that characterized it during the transition period. As has been mentioned before, during the transition process, one functional system analyst worked together with the development agile team and another one stayed as part of the external functional analysts group. The group of operational system analysts did not change.

The System Analyst Role. Following are several expressions with respect to the system analyst role as it was described by the interviewees:

- "System analyst is a person who observes a process, understands what the process needs to achieve, and checks how it is possible to improve it";
- "A process designer like buildings designer";
- "There are system analysts who will finally instruct also how to build the building";
- "[The system analyst] has a global understanding of the system, can analyze the requirements, and can connect the concepts of the operational world to technical concepts";
- "Translator from different world of concepts to a system of development concepts".

As can be seen, the role of system analyst is conceived by the interviewees as a central one, both because he or she has a wide perspective at the system and because she or he connects the different parties involved in the development process.

The System Analyst Role During the Transition Period. The interviewees also described the change they experienced during the transition process and how they tried to cope with it.

Plotkin describes two main sets of solutions to deal with phenomenon that are characterized by change, and explains how change can be coped with. None of the solutions is exclusive of the others [9, pp. 145–152].

The first set of solutions concerns with 'reducing the amount of significant change', thus reducing the change scope. One way to do it is by reducing the period of time between conception and reproductive competence; Meaning, to keep the ratio 'life-span length to numbers of offspring' low, i.e., to maintain high reproductive output in a relatively short period of time. In this case, the change is coped by keeping updated, as far as possible, the genetic instructions of each individual. Plotkin's examples in this chapter are mostly taken from animals' life. The second way to reduce the amount of significant change is to live in a relatively isolated and unpopulated place. A variation of this idea is parents' protection on their offspring by isolating them.

The second set of solutions to cope with the phenomenon of change takes the form of 'if you can't beat it, join it', i.e., change the phenotypes so that they can change with and match the changing features of the world. The first strategy is diversity. One

way to accomplish it is to *produce large numbers of different offspring* in order to increase the chance that at least some individuals will be able to face the change. The second strategy, named the 'tracking option', is to give rise to a change within phenotypes, i.e., by *producing phenotypes that change in response to changes in the world*. The tracking option is achieved by knowledge-gaining devices which, according to Plotkin, are the immune system and the intelligent mechanisms of the brain. And thus, the immune system operates in the sphere of *chemistry*, while the brain mechanisms, known as rationality or intelligence, operate in the sphere of the *physical world* of temporal and spatial relationships of events and objects.

In what follows we present some of the interviewees' expressions with respect to the change in the system analyst role during the transition process. The expressions are arranged according to the way the change is coped with.

I. Time Aspect

- "I should better understand the constraints because people start to work immediately and I immediately see their side";
- "Every two weeks I need to say what will be in the iteration";
- "There is no waste of time".

II. Place Aspect

- "Everything stays in the customer hands";
- "It helps reduce over spec".

III. Diversity

- "Everyone is involved and this raises the confidence feeling with respect to the process";
- "There is more interaction".

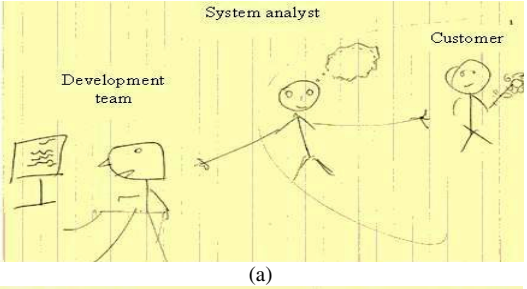
IV. Knowledge-Gaining Devices

- "XP gave us more power";
- "Sometimes we use documents and sometimes only presentations";
- "Explaining the concept, I sometimes see that my concept is wrong";
- "Process designer is like building designer".

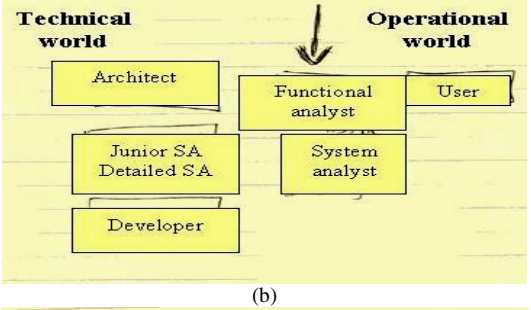
One of the most salient phenomena that were observed during the interviews was that the interviewees frequently used metaphors and even mentioned this use as a skill that may support the performance of the system analysts' role. The metaphors were diverse and come from different worlds of concepts like buildings, flowers, space ship, flow, journey, and relationship between genders.

As presented in Section 3, the interviewees were asked to draw the position of the system analyst in the organization as they see it. The three drawings (a)-(c) are presented as part of Figure 6. As can be observed, the system analyst role is mainly conceived as a bridge between the customer and the developers. Specifically, Draw (a) and Draw (c) reflect that this role holder is in a tight middleman situation; Draw (a) reflects that it is not an easy task; Draw (c) reflects also a kind of pressure between

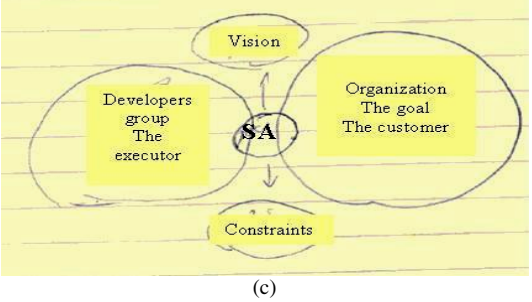
the vision and constraints. In Draw (b) the interviewee described the change that was performed in this project in which the system analyst who works on the detailed specifications tends to be part of the development group, the technical side.



(a)



(b)



(c)

Fig. 6. The position of the system analyst

We elaborate on how reflective processes can improve a person's understanding of his or her own conception. The importance of reflective processes in software engineering is presented in [5] and in [13] based on [11]. For illustration, we look at Draw (a).

Draw (a) reflects the position of the system analyst in the organization. When the interviewee was asked to reflect on his own draw, he said that the draw illustrates a conflict and that "everything is a matter of explanations". Specifically, he explains that "The customer does not know what is possible to be done. He [the customer] thinks he has a flower. He does not know that he can have two flowers". We conclude with his final words that "Sometimes there is no conflict. Sometimes it is just that customers are from Venus and developers from Mars."

5 Summary

This paper presents the process of transition to agile development in a large-scale software project in the Israeli Air Force focusing on the system analysis and design aspect. Specifically, the project specifications are compared using size and complexity measures and the change in the role of the system analysts is examined. We found that using the agile practices actually produce less complex specifications. Further, the role of the system analyst is changed in a way that improves the process and the increases collaboration.

References

1. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Reading (2000)
2. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change, 2nd edn. Addison-Wesley, Reading (2004)
3. Dubinsky, Y., Hazzan, O., Keren, A.: Introducing Extreme Programming into a Software Project at the Israeli Air Force. In: The 6th International Conference on Extreme Programming and Agile Processes in Software Engineering, UK (2005)
4. Dubinsky, Y., Talby, D., Hazzan, O., Keren, A.: Agile Metrics at the Israeli Air Force. In: Agile Conference, Denver, Colorado (2005)
5. Hazzan, O.: The reflective practitioner perspective in software engineering education. The Journal of Systems and Software 63(3), 161–171 (2002)
6. Highsmith, J.: Agile Software development Ecosystems. Addison-Wesley, Reading (2002)
7. McCabe, T.: A Complexity Measure. IEEE Transactions on Software Engineering 2(6), 308–320 (1976)
8. Mohamed, F., Schmidt, D., Johnson, R.: Building Application Frameworks. Wiley, Chichester (1999)
9. Plotkin, H.: Darwin Machines and the Nature of Knowledge. Harvard University Press, Cambridge (1997)
10. Schön, D.A.: The Reflective Practitioner: BasicBooks (1983)
11. Schön, D.A.: Educating the Reflective Practitioner: Towards a New Design for Teaching and Learning in The Profession. Jossey-Bass, San Francisco (1987)
12. Talby, D., Adler, D., Kedem, Y., Nakar, O., Danon, N., Keren, A.: The Design and Implementation of a Metadata Repository. INCOSE/IL (2002)
13. Tomayko, J., Hazzan, O.: Human Aspects of Software Engineering. Charles River Media (2004)
14. Watson, A.H., McCabe, T.J.: Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric, pp. 500–235. NIST Special Publication (1996)

Product Oriented Modelling and Interoperability Issues

Salah Baïna[1], Hervé Panetto[1], and Khalid Benali[2]

[1] CRAN (UMR 7039), University Henry Poincaré Nancy I
F 54506 Vandoeuvre les Nancy, France
salah.baina@cran.uhp-nancy.fr, herve.panetto@cran.uhp-nancy.fr
[2] LORIA (UMR 7503), Campus scientifique, BP 239
F 54506 Vandoeuvre les Nancy, France
benali@loria.fr

Abstract. The consideration of Product information or Knowledge management, product traceability or genealogy, and product life cycle management implies new strategies and approaches to manage flows of information that relate to flows of material managed in shop floor level. Moreover, throughout product lifecycle coordination needs to be established between reality in the physical world (physical view) and the virtual world handled by manufacturing information systems (informational view). This paper presents a product oriented modelling and a product oriented interoperability approach based on the use of the "Holon" modelling concept as a means for the synchronisation of both physical view and informational views. The Zachman framework is afterwards used as a guideline to establish product oriented interoperability between enterprise systems.

Keywords: Process Modelling, Manufacturing Systems, Enterprise Integration, Systems Interoperability, Model Driven Architecture, Zachman, Models Transformations and Mappings.

1 Introduction

Enterprise integration and the opening of information systems towards integrated access have been the main motivation for the interest around systems interoperability. Integration aspect and information sharing in the enterprise lead to an organisation of the hierarchy of enterprises applications where interoperability is a key issue (see Fig. 1).

This hierarchy defines the three main levels in manufacturing enterprises:

- L1: Process control level contains all processes that perform routing and physical transformations on the produced goods and services;
- L2: The Execution level performs the processes that manage decision flows (e.g.: Workflow systems) and production flows (e.g.: MES[1], SCE[2]);

[1] Manufacturing Execution System.
[2] Supply Chain Execution.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 293–308, 2008.

– L3: The management system level is responsible for the management of processes that handle all different informational aspects related to the enterprise (e.g.: APS[3], ERP[4] or CRM[5] systems).

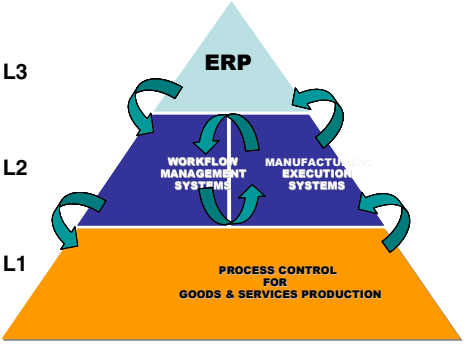


Fig. 1. Manufacturing enterprises common structure

To meet traceability, product genealogy and product life cycle management needs, nowadays an enterprise has to manage flows of information that relate to flows of material and that are managed in shop floor level. We assume that the enterprise is composed of two separated worlds (see Fig. 2):

(i) On one hand, a world in which the product is mainly seen as a physical object, this world is called the manufacturing world. It handles systems that are tightly related to the shop-floor level,

(ii) On the other hand, a world where the product is seen as a service released in the market. This world is called the business world.

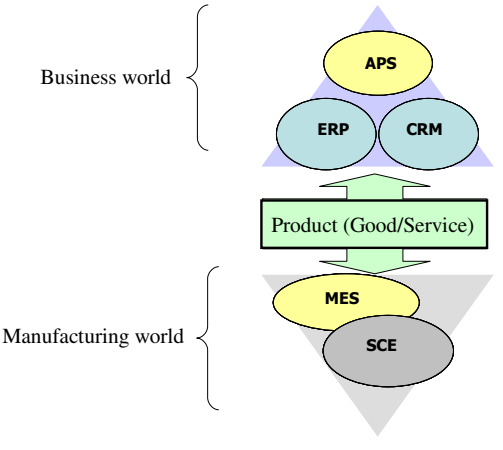


Fig. 2. Product centric approach

[3] Advanced Planning Scheduling system.
[4] Enterprise Resource Planning.
[5] Customer Relationship Management.

In order to achieve the main objective of the enterprise, "the product" to be specific, the business universe and the manufacturing universe need to exchange information and to synchronise their knowledge concerning the product (good and service). It is assumed that the product (good/service) can play the role of the gateway between both universes, since it represents a common entity between those worlds.

In this paper, we define a holon based approach in order to synchronise views in the business world and in the physical manufacturing world using the holon concept. The paper continues by presenting the usability of the concept of holon in ensuring interoperability enterprise context. Section 2 presents the bases of our holonic process modelling concepts [1] that use the product as a centric entity in process models. Section 3 of the paper gives a brief introduction to interoperability in the enterprise and explains how holons can be used as a means for enterprise applications interoperability. In Section 4, an implementation of the holon is proposed. In Section 5, we propose a guideline to model driven and product oriented interoperability based on the Zachman Framework. Section 6 gives conclusions and perspectives for this work.

2 A Modelling Construct for Product Representation

In this section, we introduce the holon as a modelling concept. Afterwards, we will show how this concept can be exploited in order to facilitate taking into account interoperability concerns in modelling phase. Existing solutions for interoperability in enterprise environment focus mainly on enterprise processes interoperability and interconnection. Throughout product lifecycle, coordination needs to be established between the reality in the physical world where the product evolves as a physical object and the "electronic" world handled by manufacturing information systems where the virtual image of the product evolves as an informational object. Our work aims to provide a product centric approach for enabling interoperability between information systems in the manufacturing environment in order to establish the coherence between the physical products and their informational representations. To take into account this duality (physical things/ informational things), we propose an adaptation of the concept of holon [2] to this specific problem.

The word Holon is a combination of the Greek word *holos*, meaning whole, and the suffix *on* meaning particle or part. A holon is an identifiable part of a system that has a unique identity, yet is made up of sub-ordinate parts and in turn is part of a larger whole. A Holon has two main features, autonomy and cooperation. Several adaptations of the holon concept have been proposed in several fields. In the manufacturing context, a Holonic Manufacturing System (HMS) is an autonomous and cooperative building block of a system for transforming, transporting, storing and/or validating information and physical objects ([3, 4]). In this paper, we adapt the *holon* concept definition to solve the problem of synchronisation between physical views and informational views of the same objects. We define the holon then as an aggregation of an information part and a physical part.

In our Holon based Process Modelling [5], holons are used to represent products; the physical part of the holon represents the material part (also called physical view) of the product and the informational part of the holon represents the informational part (informational view) of the product.

To formalize our perception of the product using the holon concept, we adapt some constructs from the well known BWW ontology to our specific purpose. Our analysis is based on the ontology initially introduced by Bunge [6, 7] and adapted by Wand & Weber for the information systems field [8, 9]. The BWW ontology has its roots in fundamental problems of conceptual modelling. Wand and Weber recognized that the quality of conceptual models is always dependent on the correspondence between the model and what the model is about.

Our fomalisation distinguish holon characteristics into two categories;

- Attributes describing the current state of the holon. The state of a holon contains three kinds of attributes: space attributes, shape attributes, and time attributes [10];
- Properties related to the holon but which do not correspond to any of the three types of properties; space, shape or time.

Holons can be classified into two categories; *(i)* elementary holons and *(ii)* composite holons:

(i) Elementary holons are the combination of a single informational part and a single physical part.

(ii) Composite holons are the result of the processing and treatment of one or more other holons, this processing can be the aggregation of a set of holons (composite or elementary) in order to compose a new holon or a transformation of one composite holon to obtain a new one.

Figure 3 represents the UML class diagram defining the holon concept meta-model. In order save place and limit the complexity, in this meta-model, we have not represented the many constraints that apply between classes and that are specified using the OCL language as defined in UML specifications [11].

Here is a brief description of this class diagram: The Class *Holon* defines basic attributes for both composite and elementary holons. A *Physical Part* is a reference to the physical part encapsulated in a *holon*. An *Elementary Holon* is defined as a holon with no indication about his lifecycle. For example a product, produced by external manufacturing systems does not give information about the processes needed for its manufacturing. A *Composite Holon* is a holon that has been obtained by either by assembling existing holons, or by disassembling existing holons into new ones.

The *state* class defines the different states that have been observed during the processing phase of the holon. Every manipulation of a holon through a process (Process Instance) implies a change in the state of the processed holon. A *Property* of a *holon* contains information that can not be handled only using its *state*. The *Process instance* refers to the execution of a process on a single holon, this class enables description of the execution of the process with high level of detail (e.g.: elapsed time, start and end of the treatment, used equipment, needed personal). A *Process instance* input is a holon state A *Process* describes an internal process that is performed inside the studied domain. The *Resource* class describes resources needed to perform a process instance. A *resource* can be a material resource, a software resource or a human resource. Each *resource* provides a set of capabilities, and each *process* needs some capabilities to be performed.

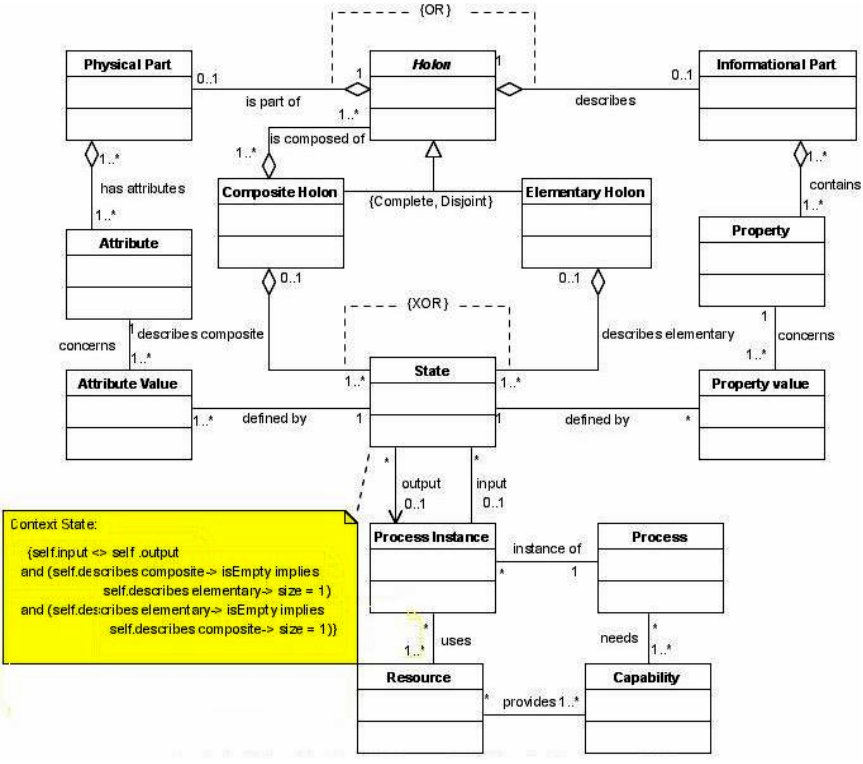


Fig. 3. Class diagram for the Holon model

3 Holon Based Models and Interoperability

The ISO/IEC 23821 Information Technology Vocabulary defines interoperability as "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units." The IEEE STD 610.122 standard defines interoperability as "the ability of two or more systems or components to exchange and use information". In this paper, interoperability definition is adapted from the two previous definitions as:

Definition 1: Interoperability is the ability to communicate, to cooperate and to exchange models between two or more applications despite differences in the implementation languages, the execution environments, or the models abstraction [12].

Interoperability can be classified into two categories considering the enterprise hierarchy model:

"Horizontal Interoperability" is the interoperability between applications from the same conceptual level in the enterprise. This first category of interoperability aims to synchronise models that were created in different enterprises even those managed by

different modelling systems (e.g.: enabling organisational interoperability between two systems used in two different organisations).

"Vertical interoperability" is the interoperability between applications from different enterprise levels. The objective of this category of interoperability is to maintain coherence between information that is handled in two different level of the enterprise (e.g.: ensuring coherence between organisational models of the enterprise and the process models used at shop floor level).

The following introduces the Levels of Conceptual Interoperability Model (LCIM). Similar to the technical approaches, five levels of interoperability are defined [13]. The focus lies on the data to be interchanged and the interface documentation, which is available. The layers are defined as follows:

Level 0 - System Specific Data: No interoperability between two systems. Data is used within each system in a proprietary way with no sharing. The component (or application) is a black box.

Level 1 – Documented Data: Data is documented using a common protocol and is accessible via interfaces. The component is a black box with an interface.

Level 2 – Aligned Static Data: Data is documented using a common reference model based on a common ontology, i.e., the meaning of the data is unambiguously described. This is also possible by using metadata standards or by using standard reference models. The component is a black box with a standard interface.

Level 3 – Aligned Dynamic Data: The use of the data within the federate/ component is well defined using standard software engineering methods such as UML. This shows the use of data within the otherwise unknown "black box behind the interface," also known as white box.

Level 4 – Harmonized Data Semantic: connections between data that are not related concerning the execution code is made obvious by documenting the conceptual model underlying the component.

In order to take into account interoperability requirements during modelling phase in the context of manufacturing systems, we introduce, in this section, the holonic modelling approach for interoperability. Existing interoperability standards and most of existing techniques that enable business process or workflow interoperability are based on a message exchange paradigm (e.g. Wf-XML, BizTalk, FIPA ACL.). These solutions resolve only the particular case of syntactic interoperability (messages vocabulary, messages format, data types, etc). In this section, we show how the holon concept can be used as a means for resolving interoperability issues. First, we will show the use of the holon to handle horizontal interoperability concerns at modelling time. Second, the case of vertical interoperability is studied.

3.1 Vertical Interoperability with the MDA Approach

In this section, we introduce an approach for interoperability based in a model driven architecture (MDA) [14]. The main objective of this section is to show how models

based on the holon concept defined in section 2 could be expressed and transformed into models based on existing data exchange standards and other unified languages.

Figure 4 shows the four-level ontological approach levels for modelling that are used in the MDA. As it is explained in [15], the lowest level M^0 presents different subjects for modelling, called universe of discourse. The level M^1 contains different models of each universe of discourse. The next level M^2 presents domain specific meta-models: one meta-model for each of the domains of interest relevant for the M^1 models. And finally, M^3 level presents a meta-meta-model designed to allow the definition of all the existing in the scope of the meta-models. In this context, applications interoperability may be solved by a top-down approach based on the four levels of the MDA. Indeed the MDA approach for interoperability relies on meta-models mapping to determine, establish and measure interoperability between applications. Several research works have been done in order to resolve meta-models mappings, more generally ontology mappings problems [16].

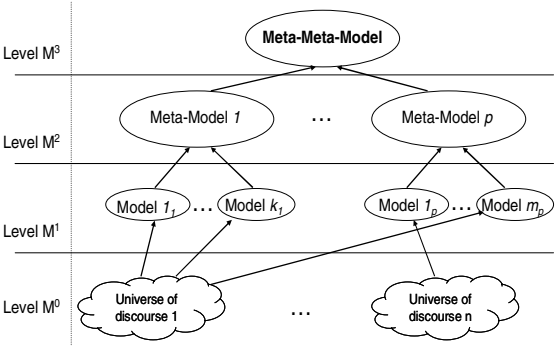


Fig. 4. The four-level ontological approach

R Lemesle, in [17], explains how models transformation can be resolved by establishing transformation rules between meta-models. Those transformation rules define a mapping that guides model transformations from the instances of the source meta-model to instances of the target meta-model. Those mappings are the bases for applications interoperability. In the MDA approach for applications interoperability, we consider that each application is based on a specific meta-model; Let us consider two applications A and B: A and B are interoperable, if and only if there is a mapping from the meta-model of A (M_A) to the meta-model of B (M_B) and a mapping form M_B to M_A. Those mappings ensure that we can build a model compatible with A from a model used by B (and vice versa).

In Order to use the MDA approach for interoperability in the holonic context, we need to define roles played by the holon in this architecture, and to position the holonic modelling approach in terms of models, meta-models and universe of discourse: M2, M1 and M0 (see Fig. 6). In the holonic context, the universe of discourse M0 concerns "The Manufacturing Enterprise Product Universe", to describe this universe of discourse we use holonic models (M1) that are instantiations of the holonic meta-model defining holons and their relationships with other entities in their environment (M2).

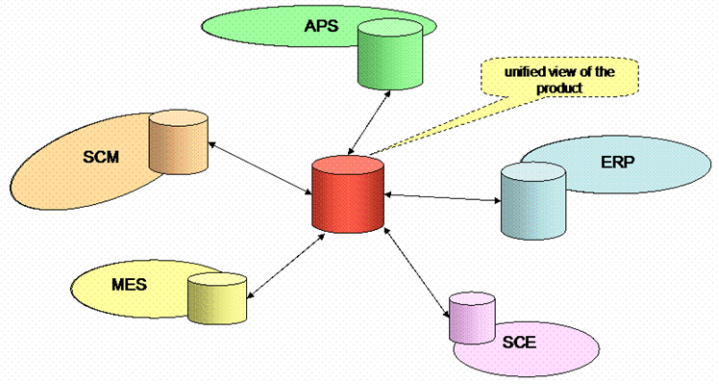


Fig. 5. A unified product view to federate enterprise information systems[6]

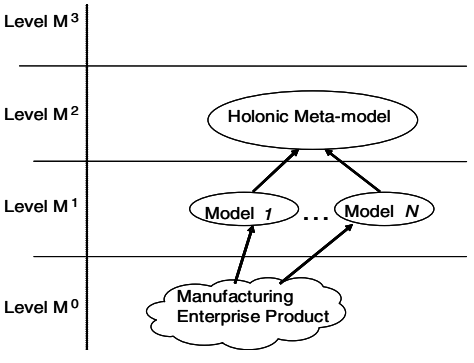


Fig. 6. Holon introduction in the MDA four ontological levels

On the bases of the conceptual model of each enterprise system (M2), transformation rules can be defined between the holon based representation of the product in on hand, and each one of applications conceptual levels on the other hand. Establishing transformation rules between those models enables exchanging product information between different enterprise applications and systems. This information exchange capacity is what we call product oriented interoperability.

Our approach for product oriented interoperability is a model driven approach based on a unified product representation (reference model), and a set of semantic transformation rules. Defining interoperability mappings between the holonic meta-model and other meta-models that handle product information enables the holonic meta-model to play the role of a reference model between those meta-models. Indeed, the holonic meta-model can be seen as a reference model for product representation.

The principle of reference model approach for interoperability is the following: there can be as many local specific models as needed (e.g. one for each system), local

[6] APS: *Advanced Planning System*; SCM: *Supply Chain Management*; SCE: *Supply Chain Execution.*

models remain as they are, and only transformation rules are defined in order to enable information synchronisation between local models and the reference model.

In the next section, an implementation of the holonic model and the interoperability mappings is proposed. This implementation relies on a commercial computer assisted software engineering (CASE) tool.

3.2 Holon Based Models for Horizontal Interoperability

Horizontal interoperability problem occurs when two or several systems or applications from the same level in the enterprise hierarchy (see figure 1) need to exchange information or data in order to perform a common objective. For example, we consider the case of a manufacturing shop-floor where several manufacturing systems need to cooperate in order to achieve a common goal, the release of the final product to be specific. In this section, we show how the use of the holon concept in the modelling phase, enables considering vertical interoperability concerns at modelling time; in the aim to facilitate resolving interoperability problems during engineering phase. To model manufacturing shop-floor, we use a minimal business process meta-model composed of four Entities:

Actor: represents a person or a group of persons that act in someway on processes or in the information system of the enterprise. An a actor can be internal or external to the enterprise

Process: is a value chain that provides a good or a service to an internal or external customer.

Site: a geographic place where the enterprise is established. Sites can express a special kind of places such as agency, office and factory, or can also express precise geographic places.

Flow: is a set of elements (data, information, energy, material ...) that are exchanged between processes

To those entities, we add the notions of *Holon* which *represents* products instances. As we see in section 2, a *holon* is described by *properties* and *attributes* that are mandatory for controlling the execution of a process on the *holon*. To manipulate those pieces of information we assume that each process is indeed composed of two interdependent sub-processes: *(i)* An informational process is responsible of manipulating, updating and controlling the information concerning the product (holon), this informational process can be implemented by an application that is performed on the information contained in the product, *(ii)* a physical process that performs all physical transformations on the material of the product. Those two sub-processes are performed in an atomic operation (both are executed or none).

Two types of relationships between a process and a piece of information (property or attribute) have been identified: production and consumption;

- *Production:* we say that a process produces an attribute (or property) when the attribute did not exist before the execution of the process;
- *Consumption:* a process is said to be consumer of an attribute (or property) when it uses the attribute (or property) or updates it.

The specification of relationships between processes and pieces of information during modelling phase enables defining the interfaces of processes at modelling time. The interface of a process defines its inputs and outputs.

Using those interfaces, interoperability of processes using can then be defined as explained in the following:

Definition 2: A process P is said interoperable with a system S (composed of processes) iff each input of P is declared as an output of one of his predecessors in S.

The precedence relation between processes is defined as following:

Definition 3: The relation of precedence is partial order between processes; we say that a process P1 precedes a process P2 ($P1 <_{Pred} P2$) if it exists a path composed of flows and processes that leads from P1 to P2. In the case cyclic systems, occurrences of execution of processes should be considered; example $P1_i <_{Pred} P2_i$ the i^{th} execution of P1 occurs before the i^{th} execution of P2.

Using the holonic modelling constructs in manufacturing context, enables the considered process interoperability to be concerned at modelling time and not during the engineering phase. This interoperability is a vertical integration of processes, since all process (informational and physical) involved in the studied system are from the same enterprise level, the process control level to be specific. The obtained interoperability is categorised into level 1 of the LCI model, it defines interfaces for shop floor process, that are seen as black boxes, since the designer does not know in advance their internal structure and characteristics.

4 Holons in Action

To experiment the holonic approach defined above in real case we have implemented this approach into a commercial CASE tool named MEGA Suite[7]. MEGA is an enterprise process modelling environment that contains a business process analysis and process modelling and design tools. MEGA has its own meta-model that described all concepts and objects ready to use in MEGA, and all relationships that exist between those concepts. This meta-model can be customized and specialised for specific users needs. MEGA Suite can be used to define, describe and exploit several kinds of diagrams (e.g: Business process Diagrams, UML Diagrams, Workflows). In our contribution, we focus only on business process diagrams; indeed they seem to be the most adequate choice for holon integration. Business Process diagrams in MEGA are based on a meta-model inspired from BPMN[8]. MEGA offers tools that enable customizing the meta-model; we used these tools to embed our own holon meta-model into the existing meta-model of MEGA in order to test the usability of our proposal.

The example presented in Figure 7 shows an example of models that can be designed using the holon modelling concept to represent products in a manufacturing process model. For the sake of simplicity, this example contains only one single process that takes a holon flow as input, and produces a holon flow as output.

[7] MEGA Suite, MEGA International, *www.mega.com*
[8] Business Process Modelling Notation, *www.bpmn.org*

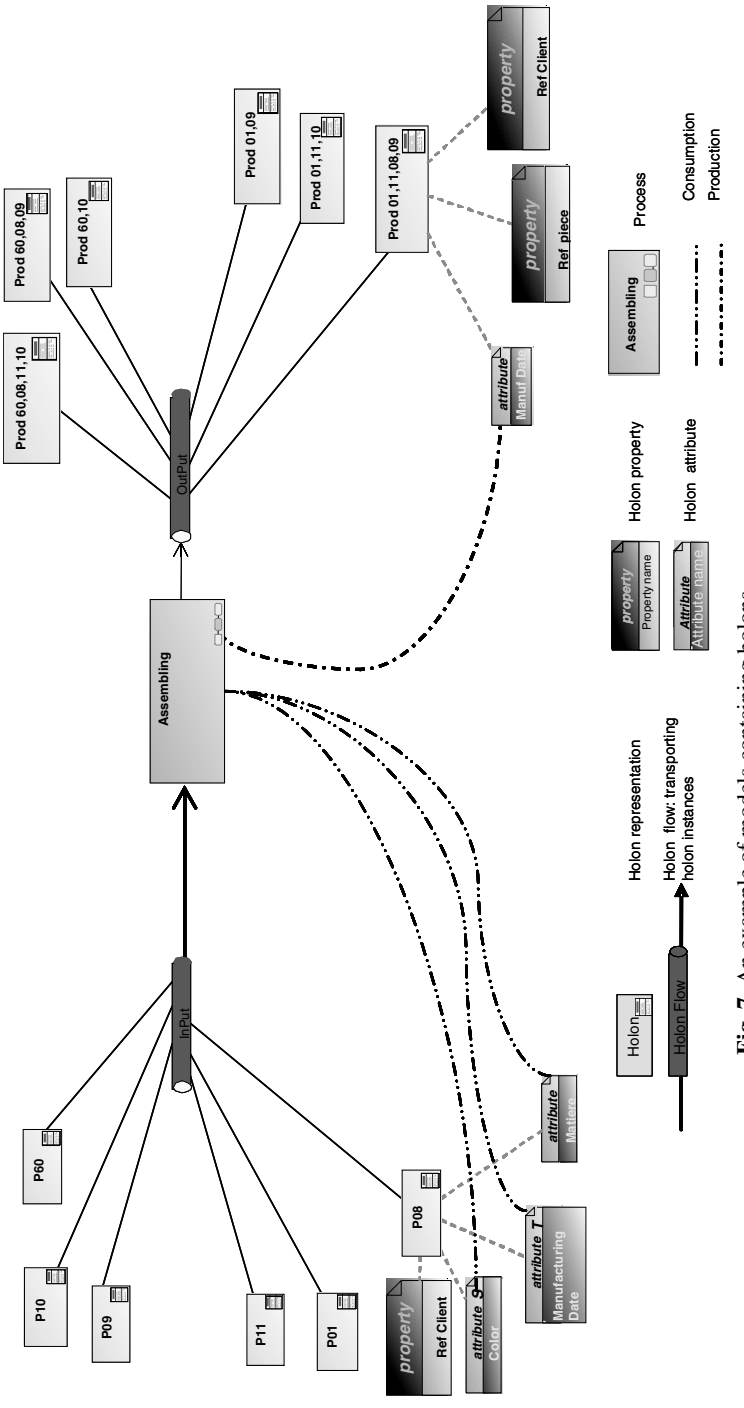


Fig. 7. An example of models containing holons

In this example, we show using the implementation of the holonic concepts in MEGA, how a process can be connected to information and data concerning holons (inputs or outputs). The holons in this example represent products (finished or not).

To experiment the holon models interoperability with other enterprise modelling frameworks using the MDA approach, two examples have been chosen; UEML and B2MML. UEML [18, 19] is the Unified Enterprise Modelling Language, it is used at the organisational level of the enterprise. B2MML [20] is an implementation of the part 1 of the IEC FDIS 62264 standard [21] developed for interfacing the manufacturing control and execution systems with higher level systems. According to the MDA interoperability approach defined in section 3, we now define an example of mappings from the holonic meta-model to The UEML and B2MML meta-models.

Mapping Holon with the Unified Enterprise Modelling Language. The Unified Enterprise Modelling Language (UEML) is the result of the UEML project [22]. The UEML is an Interlingua between Enterprise Modelling tools. The meta-model of UEML1.0 [19] defines the set of most relevant concepts and notions for Enterprise modelling.

Mapping with the B2MML language and the IEC 62264 standard. Business to Manufacturing Mark-up Language (B2MML) is an XML implementation of the IEC 62264 part 1. This standard is composed of six different parts designed for defining the models and interfaces between enterprise activities and control activities. Each model concerns a particular view of the integration problem. Those models show increasing detail level in the manufacturing system. The detail of those mappings has been published in other papers, for further information see Baïna, *et al* [5]. Vertical interoperability that is established by using those mappings is classified in the Level 2 of the LCI model. (see section 3).

To implement the mappings from the holonic models designed in MEGA and the other formats, we first define an extraction format that expresses data extracted from MEGA holon models in order to reuse it in other tools and frameworks based on other meta-models (UEML, B2MML, etc.). To represent the extracted data, we choose the XML language [23]; since it is considered as the standard application data exchange language by the W3C. MEGA Suite enables XML files generation in respect to a specific structure. XML structures for UEML [18], and B2MML [20] are used to transform the mappings defined below into XSLT rules that can be applied on the files generated by MEGA in order to restructure them into files that respects the UEML structure or the B2MML structure.

5 Model Driven Interoperability in the Zachman Framework

While well known MDA (Model Driven Architecture) [24] and MDE (Model Driven Engineering) [25] propose frameworks based on OMG specifications, highlighting different abstraction levels of a given system from application development point of view, we contribute to this domain by proposing a framework that combines various points of view of a given system linking manufacture views of a system at the “business to manufacturing” level.

Our approach provides a methodology for product information design and collection, starting from the conceptual model of the product and the manufacturing processes that interact with it, until the definition of product information data models that can be used for product information management (product quality control, traceability or genealogy). The model-driven approach proposed in this paper is derived from the Zachman framework. The Zachman Framework provides a highly structured way of defining and representing an enterprise. It uses a two dimensional classification model based around the 6 basic communication interrogatives (What, How, Where, Who, When, and Why) intersecting 6 distinct model types which relate to stakeholder groups (Visionary, Owner, Designer, Builder, Implementer and Worker) to give a holistic view of the enterprise.

	What (Data)	How (Function)	Where (Locations)	Who (People)	When (Time)	Why (Motivation)
Scope {contextual} Planner	List of things important to the business	List of processes that the business performs	List of locations in which the business operatses	List of organizations important to the business	List of events/ cycles important to the business	List of business goals/strategies
Enterprise Model {conceptual} Business Owner	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
System Model {logical} Designer	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Process Structure	e.g. Business Rule Model
Technology Model {physical} Implementer	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Detailed Representation {out-of-context} Subcontractor	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
Functioning System	e.g. Data	e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

Fig. 8. Grid representation of the Zachman framework

- How: process and functions performed in the enterprise.
- What: important data and objects of the enterprise.
- Who: human actors in the enterprise.
- Where: places, sites and locations where enterprise activity is performed.
- Why: motivations that lead business and manufacturing behaviour.
- When: events that launch activities in the enterprise.

Each cell (or artefact) in this classification must be aligned with the cells immediately above and below it. All the cells in each row also must be integrated with each other. However, cells will not be aligned diagonally.

By definition of Zachman framework, the product of a given enterprise belongs to the scope of the "What" column that describes objects that are important from the enterprise point of view. Enterprise applications and enterprise systems handle information about the product; each one of those systems has a specific representation of the product. Using retro-engineering techniques, a precise logical representation of product view handled by each system can be produced. However, a generic representation of the product is needed at the conceptual enterprise model level to unify all logical views of the product and to enable then a unified product modelling approach.

In Figure 9, holon based modelling approach are integrated into the Zachman framework in order to propose a simplified guide for product oriented interoperability based on the Zachman grid.

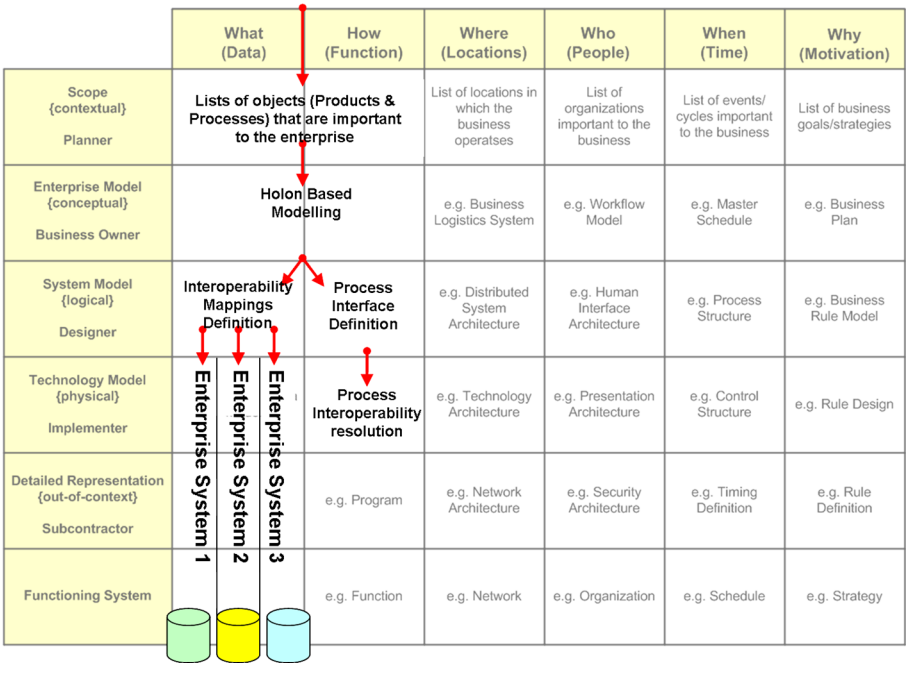


Fig. 9. Zachman framework and Holons in action for interoperability

6 Conclusion

In this paper, we defined an approach for specifying the holon modelling construct. The objective of holon based models is the synchronisation between the physical objects and their informational views in manufacturing environment. Then, we introduced how the holon approach can be used for enterprise interoperability issues. Afterwards, an implementation of our approach in a commercial CASE tool is presented. We also establish a translation mechanism based on meta-model mappings that enable applications using the holonic meta-model to exchange models

with other applications based on different meta-models, this mechanism is based on the MDA approach for interoperability.

Acknowledgements

This work was funded by the European Commission IST 6th framework programme within the framework of the Network of Excellence INTEROP. The authors would like to thank the entire INTEROP core.

References

1. Morel, G., Panetto, H., Zaremba, M.B., Mayer, F.: Manufacturing Enterprise Control and Management System Engineering: paradigms and open issues. IFAC Annual Reviews in Control 27/2, 199–209 (2003)
2. Koestler, A.: The Ghost in the Machine. Arkana, London (1967)
3. McFarlane, D., Bussmann, S.: Developments in holonic production planning and control. International Journal of Production Plannig and Control 11, 522–536 (2000)
4. Seidel, D., Mey, M.: IMS - Holonic Manufacturing Systems: Glossary of Terms. In: Seidel, D., Mey, M. (eds.) IMS - Holonic Manufacturing Systems: Strategies, vol. 1, University of Hannover, Germany (1994)
5. Baïna, S., Panetto, H., Morel, G.: A holonic approach for application interopearbility in manufacturing systems environment. In: The 16th IFAC World Congress, Prague, CZ (July 4-8, 2005)
6. Bunge, M.: Treatise on Basic Philosophy, Ontology I: The Furniture of the World, vol. 3. Reidel, Boston (1977)
7. Bunge, M.: Treatise on Basic Philosophy, Ontology II: A World of Systems. Reidel, Boston (1979)
8. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammar. Information Systems Journal 3, 217–237 (1993)
9. Wand, Y., Weber, R.: On the deep structure of information systems. Information Systems Journal 5, 203–223 (1995)
10. Panetto, H., Pétin, J.F.: Metamodelling of production systems process models using UML stereotypes. International Journal of Internet and Enterprise Management 3/2, 155–169 (2005)
11. UML, The Unified Modelling Language specifications 1.5, OMG (2003)
12. Kalfoglou, Y., Schorlemmer, M.: Formal Support for Representing and Automating Semantic Interoperability. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 45–61. Springer, Heidelberg (2004)
13. Tolk, A., Muguira, J.A.: The Levels of Conceptual Interoperability Model. In: Simulation Interoperability Workshop, Orlando, Florida, USA (2003)
14. Mellor, S.J., Kendall, S., Uhl, A., Weise, D.: Model Driven Architecture. Addison-Wesley Pub Co, Reading (2004)
15. Naumenko, A., Wegmann, A.: Two Approaches in System Modelling and Their Illustrations with MDA and RM-ODP. In: ICEIS 2003. the 5th International Conference on Enterprise Information Systems, Angers, France, pp. 398–402 (2003)
16. Kalfoglou, Y., Schorlemmer, M.: Ontology Mapping: The State of The Art, The Knowledge Engineering Review, vol. 18, pp. 1–31. Cambridge University Press, Cambridge (2003)

17. Lemesle, R.: Transformation Rules Based on Meta-Modelling. In: EDOC 1998, La Jolla, California, pp. 113–122 (1998)
18. Berio, G., Anaya, V., Boudjlida, N., Krogstie, J., Petit, M.: D3.2: Core constructs, architecture and development strategy (2003), UEML TN IST- 2001- 34229
19. Panetto, H., Berio, G., Benali, K., Boudjlida, N., Petit, M.: A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models. In: IFAC INCOM 2004 Symposium, Bahia, Brazil (2004)
20. B2MML, The World Batch Forum. Business To Manufacturing Markup Language (B2MML), version 2.0 (2003), http://www.b2mml.org
21. IEC 62264, IEC FDIS 62264-1:2002, Enterprise-control system integration, Geneva, Switzerland (2002)
22. UEML, Unified Enterprise Modelling Language (UEML) Thematic Network, in IST-2001-34229 (2002)
23. XML, Extensible Markup Language (XML) 1.0, World Wide Web Consortium 1998 (1998)
24. OMG/MDA, http://www.omg.org/mda
25. Kent, S.: Model Driven Engineering. In: IFM 2002, Third International Conference on Integrated Formal Methods, Turku, Finland (2002)

Process Modeling Value Framework

John Krogstie[1], Vibeke Dalberg[2], and Siri Moe Jensen[2]

[1] IDI, NTNU, Sem Sælandsvei 7-9 7030 Trondheim, Norway
and
SINTEF, Forskningsveien 1, 0314, Oslo, Norway
[2] DNV,Veritasveien 1, 1322 Høvik, Norway
krogstie@idi.ntnu.no,
Siri.Jensen@dnv.com, Vibeke.Dalberg@dnv.com

Abstract. This paper presents an approach to increase the value gained from enterprise modeling activities in an organization. The main objective of the approach is to facilitate communication and coordination of modeling initiatives between stakeholders and within and across projects. The approach as a normative process model is presented and discussed in the context of case projects and activities, and we conclude that although work remains both on sophistication of the approach and on validation of its general applicability and value, our results show that it addresses recognized challenges in a useful way.

Keywords: Business process modeling, enterprise modeling.

1 Introduction

Enterprises have a long history as functional organizations. The introduction of machinery in the 18th century lead to the principle of work specialization and the division of labor, and on to the need of capturing, structuring, storing and distributing information and knowledge on both the product and the business process. Business process models have for a long time provided a means to structure the enormous amount of information needed in many business processes [8]. The availability of computers provided more flexibility in information handling, and led to the adoption of modeling languages originally developed for systems modeling like IDEF0 [6]. The modeling of work processes, organizational structures and infrastructure as an approach to organizational and software development and documentation is becoming an established practice in many companies [7]. Process modeling is not done for one specific objective only, which partly explains the great diversity of approaches found in literature and practice. Five main categories for process modeling can be distinguished leading on the overviews presented in [2, 16]:

1. Human sense-making and communication to make sense of aspects of an enterprise and to support communication among different stakeholders. Sense-making models are used within an activity in order to make sense of something in an ad-hoc manner, and will usually not be maintained afterwards.
2. Computer-assisted analysis to gain knowledge about the enterprise through simulation or deduction.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 309–321, 2008.

3. Business Process Management, following up the adherence of the work process to standards and regulations. Here the model is meant to act as part of a corporate memory meant to exist as a reference point over time.
4. Model deployment and activation to integrate the model in an information system. Deployment can be manual, automatic (in automated workflow systems), or interactive.
5. Using the model as a context for a system development project, without being directly implemented (as it is in category 4).

We have investigated the practice and experience of process modeling across four business areas and a number of projects and initiatives in a large, international company. Our objective was to identify possible improvements and facilitate potential sharing of relevant resources, aiming towards an optimization of the *value* gained from modeling and models. Merriam-Webster Online defines value as: "something (as a principle or quality) intrinsically valuable or desirable". We have aimed for a company-wide, inclusive scope in our use of the term, guided by what has been deemed relevant by involved stakeholders.

Three important observations were made up front:

- Even within projects a variety of objectives were found, spanning the categories presented above. A corresponding variety was found in tools, methods and attitudes to the potential value of modeling.
- In some initiatives there were significant divergence of expectations to the modeling results and value - between different stakeholders and also over time.
- Communication and sharing of resources between projects were mainly done through more or less ad-hoc reuse of models and personnel known by project workers in advance.

From this we made three assumptions:

- Single project value and stakeholder satisfaction could be increased by to a larger degree focusing on, communicating and prioritizing between diverging expectations and objectives.
- This would require a common platform for communication about modeling initiatives expectations, objectives, and other attributes.
- Such a platform could also facilitate reuse of relevant knowledge, tools, models, methods and processes between units and projects.

These assumptions lead to the development of a framework on best practice for increasing the value of process modeling and models. This proposal consists of a taxonomy, a recommended model of activities for process modeling value increasing initiatives, and links to relevant knowledge and best practices for each step of the process. Work leading up to this work has been reported in [3,4,11,12].

The rest of this paper starts with presenting the methods used in our work, from identification of needs, development and assessment. We then give an overview of the framework of best practice for increasing the value of process modeling and models, and discuss its applicability with regard to challenges identified in earlier projects. Finally, we conclude on the applicability and usefulness within the limitations of our validation, and indicate needs for further development of the framework as well as for more large-scale validation within a wider scope.

2 Research Method

The research presented in this paper is based on qualitative analysis of a limited number of case studies. According to Benbasat, Goldstein, and Mead [1], a case study is an approach well suited when the context of investigation takes place over time, is a complex process involving multiple actors, and is influenced by events that happen unexpectedly. Our situation satisfies these criteria, and the work has taken place within the frames of a three year project, including one in-depth case study, and several other less extensive case-studies. In deciding whether to use case studies or not, Yin [15] states that a single case study is relevant when the goal is to identify new and previously not researched issues. When the intent is to build and test a theory, a multiple case study should be designed. The intention of our study has been to find out how to increase the value of modeling and models in an organization. There has not been reported much research within this area previously, and we have therefore chosen a multiple case approach for the work presented in this paper.

The framework for increasing value of process modeling and models has been developed through an iterative process, refining the model. We can identify four iterations.

1. In the first iteration we studied the modeling initiative in a particular project in detail, using observation, participation, and semi-structured interviews. After a phase of explorative research, we focused on identifying the expectations and experiences towards the modeling and the models, on their score related to process modeling success factors, as well the extensive reuse of the models across the organization, viewing this as possible knowledge creation and sharing as a part of organizational learning. A hypothesis on process modeling value was established.
2. In the second iteration, we went through semi-structured interviews with representatives of several different modeling initiatives throughout the organization to survey their experience with modeling, especially with respect to benefits and value of reusing knowledge through models across projects and organizational units. A number of initiatives were selected for the study where we were able to get in-depth knowledge from those involved in the process. An interview guide was established. These interviews were focused on expected and experienced use and value from the modeling efforts in the case study, aiming at identifying as many expectations as possible, including any that may not have been documented in project documentation, because they were not considered directly relevant to the primary goals of the project. Following open questions, the interviews were structured around keywords from the work of [14] concerning “process modeling success”. Results of the study are based on these interviews, studies of project documentation and models. The interview guides were used as basis for structuring contact summary sheets with the main concepts, themes, issues and questions relating to the contact [9].
3. As a third iteration we carried out a workshop with a group of modeling experts, discussing the framework in relation to their own experiences through numerous process modeling projects. This resulted in an updated version of the framework.
4. In the last iteration, we included the framework in an actual business project using action research, where one of our researchers also acted as a modeler. This was an informal test of the framework, but gave valuable input to validate and update it.

Our results this far have certain limitations relative to internal validity [9] as representatives of some of the involved roles have been followed more closely than others. As for descriptive validity (what happened in specific situations) the close day to day interaction with the users, especially in the first and the last iteration by one of the researchers, give us confidence in the results on this point. As for the interpretive validity (what it means to the people involved) we have again in-depth accounts from central stakeholders, but again not all stakeholders have been involved to the same degree. The same can be said on evaluative validity (judgments of the worth and value of actions and meaning). That we find many results that fit the categories of existing theoretical frameworks gives us confidence on the theoretical validity of the results.

3 A Framework for Increasing the Value of Process Modeling

This best practice framework aims to increase the value of the modeling and models through enhanced awareness about current and future stakeholders, any (potential) conflicts of interest, stakeholder expectations and potential value to be gained, as well as any negative effects increasing total cost. Based on this knowledge, decisions regarding resource allocation, choice of modeling methods and tools, delegation of responsibilities etc. can be made to optimize the value of a modeling activity and its resulting models, on a project level as well as on an organizational level. The basic elements of the framework are a recommended main *process* (see Fig. 2) and some basic *concepts* (see Fig.1), elaborated on in the description of each step in the main process.

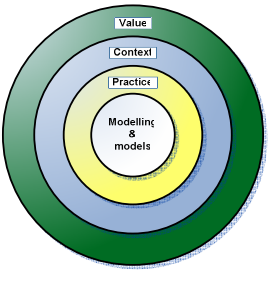


Fig. 1. Main concepts of value framework

Context is the surroundings of an initiative that might influence decisions. *Value* is identified in relation to the identified context, but also on potential value outside the initial project scope. The *practice* focuses on the strategies and practice around the modeling and the models.

The recommended process is initiated when a need for modeling has been identified. It contains four areas. Its three main steps related to the individual projects are detailed below. 'Manage modeling capabilities' is work done at an organizational level.

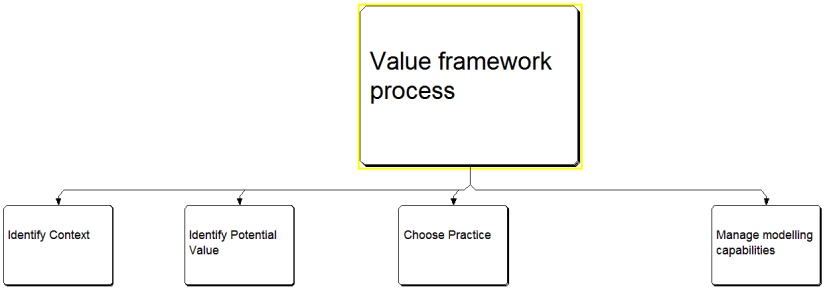


Fig. 2. The overall framework

3.1 Identifying Context

Identifying the context is mostly about expressing the circumstances of the identified need for modeling, as a basis for further communication, prioritization and planning. It will usually coincide with the writing of an application for funding, development of a project mandate and/or a project plan. At this step one should keep within the scope of the initial need, usually expressed in traditional project documentation with formal obligations. The main issues to be clarified are detailed in Fig 3, and include:

- Identification of the organizations installed base, including existing reusable models or descriptions and other relevant tacit or explicit constraints.
- Identification of the context of the modeling or model activity/initiative, including users and other stakeholders, uses, and objectives.

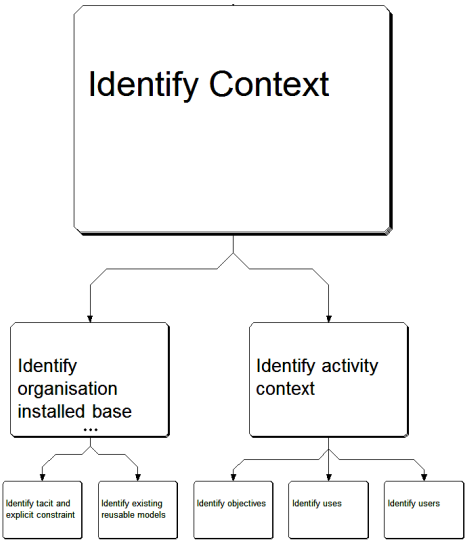


Fig. 3. Identify context

There are different actors related to a modeling initiative and a model, holding one or more *roles*. *Users* are using the models or participating personally in the modeling in order to achieve objectives. Other *stakeholders* may not be using the models directly, but extract value from planned objectives. Techniques e.g. from user-centered design is useful at this stage in the identification of stakeholder types. *Use* includes how the modeling and models are going to be used in order to achieve the objectives. *Objectives* are the goals and purposes of the modeling and models. *Installed base* includes tacit and explicit assets already existing in the organization that will have influence on the modeling and model context. *Constraints* include issues such as personal and organizational knowledge, which may be tacit or explicitly expressed constraints, organizational guidelines or instructions (explicit constraints), existing tools and languages etc. *Reusable models* are models or other documentation that were created for other purposes, but that could be reused in the new project.

Table 1. Examples of activity context

Uses	Users	Objectives
• Facilitate human understanding and learning • Communication tool • Support process improvement • Support process management • Support the work process • Automate process guidance • Automate execution support • Computer-assisted analysis • Model deployment and activation • Basis and context for software development • Training of personnel • Measuring and analyzing processes • Document best practice • Requirement specification tool	• Top management • Middle management • Work executers • Software process engineers • Project managers • Software engineers • System engineers • Software developers • Software executives • Customer management	• Increase human understanding • Improve process • Manage process • Work more effectively • Improve business performance • Less faults • Common view of future directions

3.2 Identifying Potential Value

In step 1, we identified the context where the modeling and the models were meant to play a role. In step 2, *"Identify potential value"*, the aim is to capture any (potential) extra and positive benefits of the modeling and models, exceeding the primary objectives captured in step 1. Value may be connected to the resulting models, or to the modeling activity in itself.

Often the objectives identified in step 1 will relate to the concrete modeling project, while any potential value to the rest of the organization will typically be ignored in the formal project documentation developed at this stage – due to a lack of awareness, or to avoid complicating responsibilities and bindings.

Value can be explicit and easy to grasp, but also tacit. Tacit value, e.g. the improved understanding of a work process for a modeler originally producing models for others,

are often not explicitly captured in traditional project documentation, but may still affect decisions before or during a project, or the perceived value of the project in retrospect. Future reuse of the models can be an added value of the current modeling and models, especially if this potential is taken into account at an early stage.

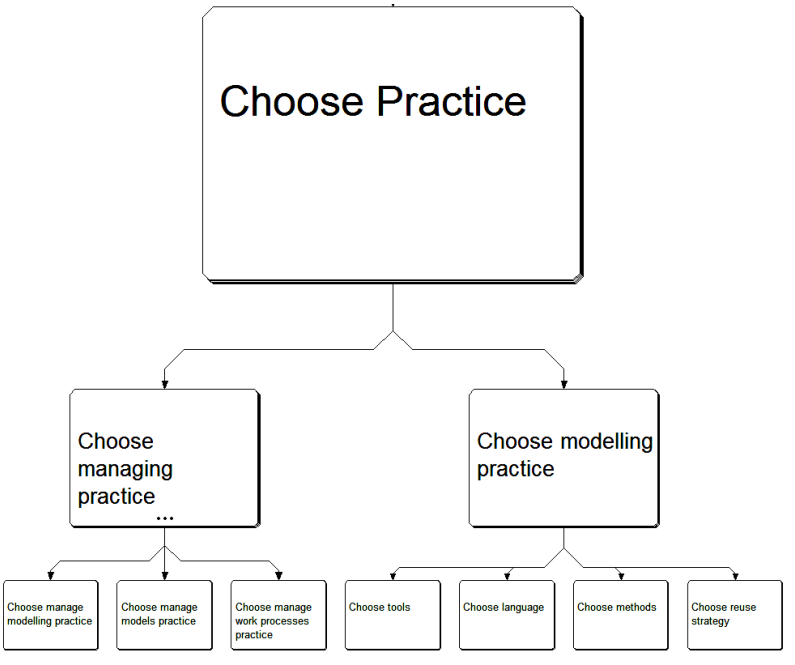


Fig. 4. Choose practice

3.3 Choosing Practice

The choice of a suitable practice should be based on the identified contexts of the modeling and models, as well as the identified expected value. Modeling practice include reuse strategy, methods, languages and tools, while managing practice define

Table 2. Examples of modelling practice

Reuse strategy	Method	Language	Tool
• Do not reuse anything, start from scratch • Reuse documentation • Reuse models indirectly or directly • Reuse methods, language and/or tools	• Visual representation • Modeling • Mapping • Text description • Modeling workshops	• IDEF0 • UML • "Boxes and arrows" • Text • Checklists • Templates • Images • Video clips	• Visio • PowerPoint • METIS • Aris • Excel • Word • "Pen and paper"

Table 3. Examples of managing practice

Manage modelling	Manage models	Manage work processes
• Manage the methods, languages and tools • Keep the models updated	• Place the models on intranet	• Instruct the modeler when and how the work processes should be updated

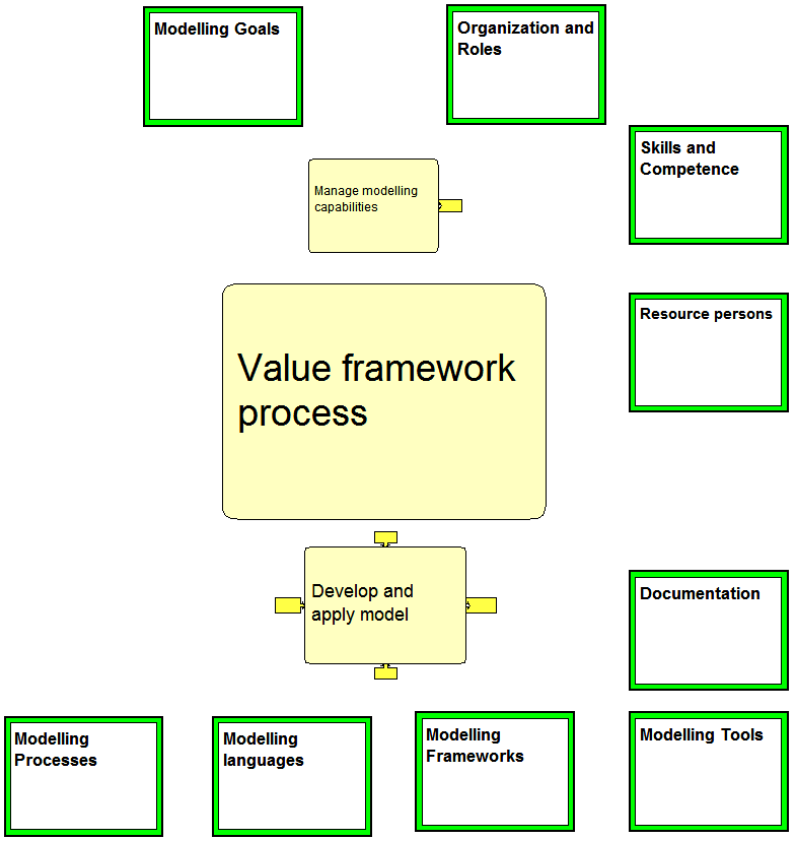


Fig. 5. Top level METIS model for structuring resources related to the framework

how to manage the modeling, the models and the work processes. The general framework of quality of models and modeling languages inspired by organizational semiotics [10] is especially helpful here relative to modeling practice related to methods, languages, and tools, having the stakeholders of the models and the goals of modeling already defined in the previous steps. When goals or stakeholder types are changed during a modeling project, one needs to reassess these aspects, and potentially select a new modeling language, method or tool.

The choice of *modeling practice* (Fig. 4) includes deciding what methods, languages and tools to be used for the development of the models. The need of formality

may differ based on the context and the expected value identified. Modeling for corporate memory may require more formality in the methods, tools and languages, than modeling for sense-making, where the model serves as an artifact for a limited time.

It is important to recognize the differences of the three areas of modeling, models, and work processes, and to assign these to three different roles. The *practice of managing* the modeling process, the models and the work processes being modeled has to be based on the identified contexts and potential values, and on the chosen modeling practice. If the context is pure sense-making, and the models are to be thrown away after they are modeled, an establishment of a practice for managing of model and work process is not necessary. The closer the context of use is to corporate memory, the more important is it to have a formal managing practice. Dependent on the contexts of use and the modeling practice, it has to be decided whether the models and the work processes should be managed after the modeling activity or not. It is important to differ between the management of the work processes itself and the models.

3.4 Framework Model

To increase the likelihood of dissemination, we have developed a model of the framework and related areas in the METIS modeling tool [13]. Fig. 5 is a screen shot of the top-level of this model, indicating the areas of information provided (not showing the internal links between different areas).

4 Applying the Framework

Through the cases we have identified expected and experienced value of modeling work and models, as well as experienced challenges. In this chapter we quote some of the reported (potential) value. We will then look into how the framework addresses the reported challenges.

4.1 Identifying Potential Value

The stakeholders in our case studies indicated many valuable outputs in addition to those initially intended for the modeling project. Some of these are:

Communication:

- The high-level models encouraged an agreement among the management participants that was vital for the rest of the project, creating important common references, identification and enthusiasm.
- The models triggered communication, being something that everyone could relate to. “Three boxes and some arrows: This is a fantastic communication tool”.
- Communication was initiated and facilitated by and through the models.
- Modeling is seen as a mechanism to extract knowledge from people’s heads.

Learning:

- The modeling process itself turned out to be a learning experience for the participating domain experts, increasing their knowledge about the processes.

- Through the workshop sessions the participants learned a lot from interacting with each other, "new" information was uncovered, and understanding improved.
- People understand themselves better after a modeling session.
- The participation in the modeling process of domain experts is important. The result would not have been the same if modelers from outside created the models based on interviews.
- The models helped taking care of and storing the *competence* of people in the organization.
- Training takes less time when process models were used.

Long-term benefits:

- The process model gives the organization one language and one tool for everyone in the organization; a common frame of reference.
- Simple and effective diagrams show what is important for the organization.
- Through modeling AsIs (the current situation), and not only ToBe (the requested situation), best practice is secured and not forgotten.
- The models are used in *marketing* towards potential customers. There is a marketing value in telling the world that they have documented processes.

4.2 Addressing Challenges of Modeling

In order to extract more value from the modeling initiatives and the models, we will in the following address some of the major identified challenges in our case studies, and examine how the framework could indicate a solution to these. For each paragraph we state the challenge, then how it is addressed in the framework.

Challenge 1: During organizational changes, models may have to be merged as processes are unified. Different modeling tools and languages increase the challenge.

Example: Several as-is processes were to be harmonized and their documenting models merged into one common process model. The models were created for different user groups, originated in different organizational units and also countries. The modeling processes were also different, involving different types of people.

Framework application: Such models are most likely based on different methods, languages and tools, created for different objectives, uses and users and other stakeholders. The historic context and the organizational installed base of modeling and managing practice of each of the models should be investigated in order to establish a re-use strategy and choose the correct current modeling and managing practice.

Challenge 2: To handle situations when the modeling starts out as an informal activity, but the resulting models develop into a process defining tool in the BPM sense. The original language and tools often do not meet new expectations for the model to be kept updated, be scaleable, and extendable with new functionality. The experience is that the chosen tool and language often do not fit into this new scenario.

Framework application: Awareness of where on the scale of sense-making versus corporate memory the models were initially created, and where on the scale the models have ended up (and where they can be expected to end up). Sense-making models

do not require a very high level of formality, while corporate memory models often do. Being conscious about this will make it easier to identify what has to be changed in the modeling and managing practice in order to align with the new situation.

Challenge 3: To keep the models and other descriptions updated and consistent.

Example: It becomes difficult to keep the models updated as the complexity increase, and the number of non-integrated tools increases.

Framework application: The framework suggests careful analysis of the expected model context before choosing the modeling practice. Considering the future complexity when choosing methods, language and tools will make model management easier. The framework also states the importance of viewing the management of the models as a specific activity, stressing the importance of appointing a model responsible. This is a different role than the modeling responsible or the work process responsible (process owner).

Challenge 4: To implement the models in the organization, particularly outside the modeling team.

Example: It is often a challenge to make the models an integrated part of the organization, and to involve the users to the extent that they feel an ownership and responsibility for them. When the persons doing the modeling leaves the project and the modeling is left to the domain experts to finish, implement and keep updated, experience shows that the focus on the models often fades. If the modelers leave too early, the models may not be implemented.

Framework application: Identify all the expected users and other stakeholders during the initial phase of the modeling activity, look into their expected areas of use and identify potential value. By choosing a modeling practice to increase the value across all identified stakeholders, ownership and usefulness is improved even for stakeholders not participating in the modeling. If many stakeholders should be involved in the modeling one can use techniques such as ”modeling conferences” [5].

Challenge 5: To produce views of the model according to different needs.

Example: Specific users and specific objectives of use require adapted views of the model. The creation of these is a challenge, both technically and as regards content.

Framework application: Identify the users and other stakeholders as parts of the context, analyze their background knowledge and needs, and what each of them are going to use the models for. Methods, language and tools should then be chosen based on this.

Challenge 6: The models often restrict and limit the communication.

Example: High level models are easy to agree upon, but real gaps between the model and current situation stay uncovered. A model is only one view of the world. When a model is the communication generating artifact, the discussions often leave out those issues not included in the model.

Framework application: Carefully identify the context and the potential value of the modeling and models before creating the models. Consciousness about how to increase

the potential value of communication will potentially help creating a more fitting model. Awareness of the limitations of a model and its restrictions is the key.

Challenge 7: The models are used in situations they were not intended for.

Example: Models are often created primarily for one objective. This is challenging when others want to use them as basis for other work, especially if the original assumptions are not documented.

Framework application: Through an analysis in the early phase of the modeling activity, identify the primary use as well as potential future use and additional potential value. Accommodation of indications of future use of the models should be considered when choosing the modeling and the managing practice. When in a re-use situation, where a modeling initiative is going to re-use earlier developed models, it is important to investigate the context the models were created for, and what modeling and managing practice have been used. The decision of a re-use strategy should be based on this investigation.

Challenge 8: To be conscious about distributing the responsibility of the modeling, models and processes correctly.

Example: One person was responsible for everything that had to do with the processes *and* the models.

Framework application: The framework makes distinctions between the activities of managing the modeling, the models, and the work processes. One role is related to the management of the modeling, another to the management of the models, a third to the management of the work processes.

5 Conclusion and Further Work

Based on extensive research across units and projects in an international company, we have identified expectations, challenges and experience pointing to potential increase in value from modeling activities. To support the realization of these values, a Process Modeling Value Framework has been developed.

The Value Framework has been evaluated against challenges and experiences of earlier modeling initiatives, as well as tested in a modeling project. There are clear indications that further development and use of the framework will facilitate communication and alignment within and between project initiatives and organizational units, thus potentially increasing value from projects through improved relevance and quality of results as well as reduced cost.

Our research has been practically oriented, aiming towards identification of the important issues in real-life modeling projects and activities, both with regard to the actors' motivation and their experience. Based on the broad investigations we have made, we are confident that our results are valid for the case company.

We expect our findings to be reproducible for other enterprises of similar size and complexity, but this still remains to be shown.

Even within the presented enterprise, on a practical level, there is still a way to go to implement and collect real-life experience with the framework. Our studies demonstrate

feasibility and advantages of use, but do not address the actual adoption of the framework by practitioners not involved in the development.

We have identified advantages both on a project and organizational level, and we expect that the project level advantages will be sufficient to motivate for the use of the framework – and that the organizational level advantages can be realized this way. This assumption however still has to be tested – and a successful implementation in the whole organization will, as a minimum, require a dedicated dissemination and marketing effort.

References

1. Benbasat, I., Goldstein, D.K., Mead, M.: The Case Research Strategy in Studies of Informations Systems. MIS Quarterly 11(3), 369–386 (1987)
2. Curtis, B., Kellner, M., Over, J.: Process Modelling. Communication of the ACM 35(9), 75–90 (1992)
3. Dalberg, V., Jensen, S.M., Krogstie, J.: Modelling for organisational knowledge creation and sharing. In: NOKOBIT 2003, Oslo, Norway (2005)
4. Dalberg, V., Jensen, S.M., Krogstie, J.: Increasing the Value of Process Modelling and Models. In: NOKOBIT 2005, Oslo, Norway (2005)
5. Gjersvik, R., Krogstie, J., Følstad, A.: Participatory Development of Enterprise Process Models. In: Krogstie, J., Siau, K., Halpin, T. (eds.) Information Modeling Methods and Methodologies, Idea Group Publishers, USA (2004)
6. IDEF-0: Federal Information Processing Standards Publication 183, Announcing the Standard for Integration Definition For Function Modelling (1993)
7. Iden, J., Eikebrokk, T., Olsen, D.H., Opdahl, A.L.: Process Change Projects: A Study of Norwegian Practice. In: European Conference on Information Systems, Gothenburg, Sweden (2006)
8. Hammer, M.: Reengineering Work, Don't automate, Obliterate. Harvard Business Review (1990)
9. Miles, M.B., Huberman, A.M.: Qualitative Data Analysis, SAGE Publications (1994)
10. Krogstie, J., Sølvberg, A.: Information systems engineering - Conceptual modeling in a quality perspective. Kompendiumforlaget, Trondheim, Norway (2003)
11. Krogstie, J., Dalberg, V., Jensen, S.M.: Harmonising Business Processes of Collaborative Networked Organisations Using Process Modelling. In: PROVE 2004, Toulouse, France (2004)
12. Krogstie, J., Dalberg, V., Jensen, S.M.: Using a Model Quality Framework for Requirements Specification of an Enterprise Modeling Language. In: Siau, K. (ed.) Advanced Topics in Database Research, vol. 4, Idea Group Publishers, USA (2005)
13. Lillehagen, F.: Visual Extended Enterprise Engineering Embedding Knowledge Management, Systems Engineering and Work Execution. In: IEMC 1999. IFIP International Enterprise Modelling Conference, Verdal, Norway (1999)
14. Sedera, W., Rosemann, M., Doebeli, G.: A Process Modelling Success Model: Insights From A Case Study. In: 11th European Conference on Information Systems, Naples, Italy (2003)
15. Yin, R.: Case study Research. SAGE Publications (1994)
16. Vernadat, F.: Enterprise Modelling and Integration. Chapman and Hall, Sydney, Australia (1996)

Combining SDBC and ISDL in the Modeling and Refinement of Business Processes

Boris Shishkov[1] and Dick Quartel[2]

[1] University of Twente, Department of Computer Science, Drienerlolaan 5
7500 AE Enschede, The Netherlands
[2] Telematica Instituut, P.O. Box 589, 7500 AN Enschede, The Netherlands
b.b.shishkov@ewi.utwente.nl, dick.quartel@telin.nl

Abstract. Focusing on the alignment between business requirements and application (software) functionality, the SDBC approach considers three viewpoints complementing each other, namely statics, dynamics, and information. Next to that, the approach addresses systematically and separately business modeling and application design, applying the mentioned viewpoints to both of them. The approach also allows for an adequate extension of its 'dynamic' business modeling, acknowledging the real-life complexity that includes communication and coordination issues, such as meanings, intentions, commitments, and obligations. Hence, in order to consider appropriately these (communication and coordination – related) issues as complementing its dynamic business modeling, SDBC applies at least two modeling techniques. The transformation between them nevertheless complicates the modeling process; furthermore, different techniques use different modeling formalisms whose reflection sometimes causes limitations. For this reason, we explore in the current paper the value which the modeling language ISDL (allowing for useful refinement of business process models) could bring to SDBC, particularly in the elaboration of dynamic (behavioral) business models with real-life aspects. We also explore how SDBC can benefit from ISDL-related methods assessing whether a realized refinement conforms to the original process model. The results reported in this work are usefully supported by an illustrative example.

Keywords: System design, Business process modeling, Refinement, SDBC, ISDL.

1 Introduction

A number of software development approaches have failed because of being insufficiently capable to grasp and utilize the original business information. As claimed in [16], the specification of software and the analysis/modeling of its corresponding business processes, should be considered as one integrated task.

The ***SDBC*** ('SDBC' stands for *Software Derived from Business Components*) approach [13,14,15,16] addresses this challenge, by allowing for a sound *mapping*

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 322–335, 2008.

between a business process model and a software specification model. They both are approached through different complementing *viewpoints*, the consistency among which is certainly crucial [4].

SDBC considers three essential modeling viewpoints, namely: *statics* (about the static relationships among entities), *dynamics* (about behavior), and *information* (about data). Next to that, the approach addresses systematically and separately business modeling and application design, applying the mentioned viewpoints to both of them. The approach also allows for an adequate extension of its behavioral business modeling, acknowledging the real-life complexity that includes *communication and coordination issues*, such as meanings, intentions, commitments, and obligations. These could usefully be reflected in another (complementary) viewpoint, namely *communication viewpoint* (as in the SDBC terminology), which plays a role with respect to real-life semantics and pragmatics [14].

Thus, in SDBC the behavior viewpoint and the communication viewpoint are considered in combination, as complementing each other. In applying SDBC, for example, one could firstly use the *DEMO Process step model* [3] for capturing meanings, intentions, commitments, obligations, and so on, and secondly - reflect this in a *Petri Net business process model* [1]. Hence, in order to combine properly these two viewpoints, SDBC would have to use at least two modeling techniques. The transformation between them nevertheless complicates the modeling process; furthermore, different techniques use different modeling formalisms whose reflection sometimes causes limitations.

Since the modeling language ***ISDL*** [6,8] - 'ISDL' stands for *Interaction Systems Design Language*, is powerful as it concerns the refinement of business process models and corresponding assessment for correctness, it is feasible to expect that ISDL can usefully complement SDBC, by allowing refinement of business process models, from the perspective of communication and coordination.

This has motivated our studying potentials for combining SDBC and an integrated modeling facility based on ISDL. In particular, we explore in the current paper the value which this modeling facility could bring to SDBC, particularly in the elaboration of behavioral (dynamic) business models with real-life aspects. We also explore how SDBC can benefit from ISDL-related methods assessing whether a realized refinement conforms to the original process model; actually, the existence of such ISDL-related conformance assessment methods further justifies the claim that ISDL can be useful in the refinement of SDBC dynamic business models. ISDL can also add value in the SDBC-driven mapping of such models towards software specification, particularly in the context of the design of software services [9], since a mapping mechanisms exist between ISDL and BPEL/WSDL specifications.

The outline of this paper is as follows: Section 2 considers SDBC, paying particular attention to concepts that concern the dynamic (behavior) and communication viewpoints. Then we present in Section 3 an illustrative example to be used in our further studies. On this basis, we discuss in Section 4 how SDBC and ISDL can be usefully combined in the modeling and refinement of business processes. Section 5 then analyzes the value of applying SDBC and ISDL in combination. And finally, Section 6 presents the conclusions.

2 SDBC

As suggested in the Introduction, SDBC is envisioned to be a useful modeling framework that approaches business processes from the perspective of related software-specification. This claim has been motivated in [13], where relevant strengths of SDBC are justified: (i) business process modeling based on the theories of LAP and OS [14]; (ii) component-based business-software alignment; (iii) re-use of modeling constructs; (iv) software specification consistent with the Unified Modeling Language – UML [12]. However, the problem addressed in this paper is the business process modeling consistency, particularly with regard to the dynamic and communication viewpoints. Thus, in this section, we firstly outline SDBC, and we secondly consider SDBC-related concepts that concern the mentioned problem.

2.1 Outline and Relevant Features

In summarizing SDBC, we use the following abbreviations as applied in Figure 1: bc – *business component* (a business sub-system that comprises exactly one business process); bk – *business coMponent* (a model of a business component, which is elaborated in terms of statics, dynamics, data); glbk – *general business coMponent* (which is re-usable by extension); gcbk – *generic business coMponent* (re-usable by parameterization); ssm – *software specification model*; sc – *software component* (an implemented piece of software representing a part of an application); sk – *software coMponent* (a conceptual specification model of a software component). For more information on the above concepts interested readers are referred to [13].

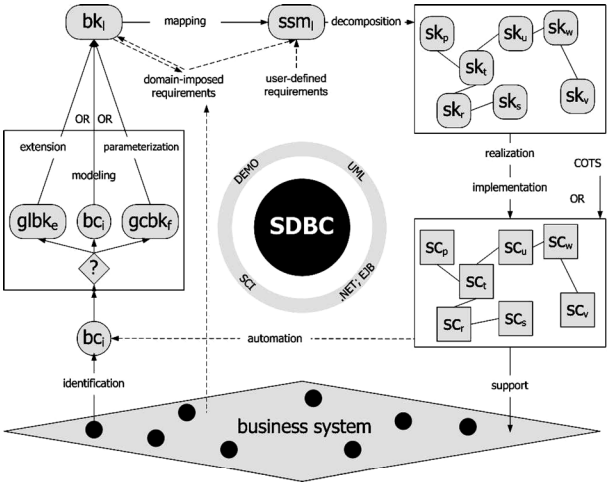


Fig. 1. Outline of the SDBC approach [13]

The figure shows that SDBC is about a component-based business-process-modeling-driven specification and realization of software. The starting point is the consideration of a business system that might be identified and elicited either by using

a scenario or by an abstract business modeling input [14]. Business components are then identified (denoted with textual descriptions), by applying the *Semantic Analysis Method* – SAM leading to the derivation of the so called '*SCI modeling output*' [7,13]. They are then reflected in corresponding business coMponents, in supplying an adequate modeling foundation for the further software specification activities. Another way of arriving at a business coMponent is by applying re-use: either extending a general business coMponent or parameterizing a generic business coMponent. DEMO and other *Language-Action-Perspective*-(LAP)-driven modeling tools [18] are relevant as far as business coMponents' specification is concerned. Each business coMponent should then be elaborated with the domain-imposed requirements, for the purpose of adding elicitation on the particular context in which its corresponding business component exists within the business system. Then, a mapping towards a software specification model should take place, possibly driven by the DEMO-UML transformation mechanism introduced in [17]. The *domain-imposed requirements* as well as the *user-defined requirements* are to be considered here, since the derived software model should reflect not only the original business features but also the particular user demands towards the software system. The (UML-based) software specification model would then need a precise elaboration, achieved partially through its decomposition into a number of software coMponents reflecting functionality pieces [15]. Then these software coMponents are to undergo realization and implementation, being reflected (in this way) in software components. This final set of components might consist of such components which are implemented (using software component technologies, such as .NET and EJB) based on corresponding software coMponents and such components which are purchased. The resulting component-based application would support the target business system, by automating anything that concerns the initially identified business component(s).

SDBC is thus not only capable to adequately capture semantic and pragmatic real-life aspects but it can also support their further mapping towards software specification, consistently with the *de facto* standard, UML. The SDBC business-software alignment itself is component-based, founded in the *CBD paradigm* – 'CBD' stands for *Component-Based Development* [13]. Such an alignment allows for good traceability between business and software modeling constructs. Finally, the component-based business-software alignment allows for re-use of modeling constructs. This essentially improves the modeling process since building new models includes the re-use of previously built modeling constructs.

2.2 Concepts

SDBC addresses the communication viewpoint, by applying the LAP theory, providing an innovative interpretation of the LAP-driven *transaction* concept.

The generic process of a transaction is depicted in Figure 2. If everything goes smoothly, a *request* is followed by a *promise*, which is followed by a *statement* (preceded by a non-communicative *production act*) which is followed by the *acceptance* of the production fact. However, an entity could also enter discussions (negotiations). For example, if Mary asks for a pizza, it might happen that the sales-person (Paul) says that the shop is closing soon and only hamburgers could be offered – so, this is the discussion, Mary could accept this or not. If she accepts, Paul states a

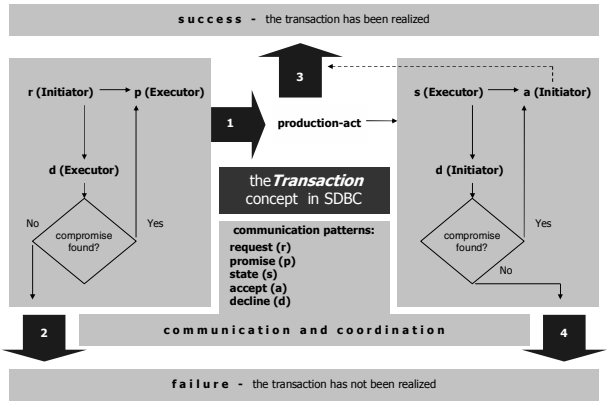


Fig. 2. The transaction concept in SDBC

promise regarding this updated request. Next, if she does not like the hamburger, when Paul states it is ready, they again enter a discussion (whether another hamburger should be delivered or the money – returned back, for example). Depending on the outcome of such discussions, a transaction could reach failure and no *production fact* would then have appeared. That is why Figure 2 presents success and failure 'layers'.

Hence, we have four possible communication outcomes concerning the *initiator* of the transaction (Mary, in the example) and its *executor* (Paul), as shown on the figure: 1(2) – agreement is (not) reached and the executor will (not) realize a production act; 3(4) – the initiator has (not) accepted the delivered result and a Transaction has (not) appeared.

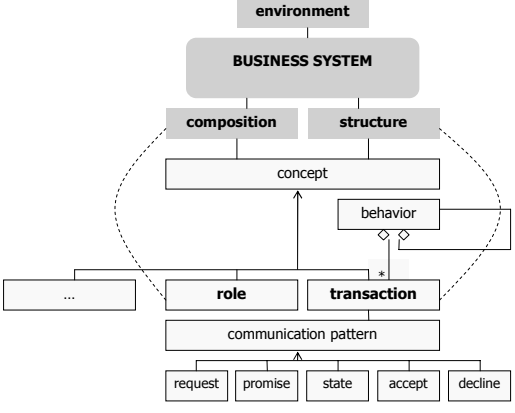


Fig. 3. SDBC concepts

Thus, the elementary business process modeling building blocks in SDBC are transactions; we consider the communication patterns (discussed above), namely: **request**, **promise**, **state**, **accept**, and **decline**, needed for the elaboration of a

transaction. Furthermore, by adopting a subjectivist philosophical stance, SDBC acknowledges that nothing exists without a perceiving/acting agent [7], and especially addresses the entities related to corresponding transactions. However, instead of considering the particular agent (entity) involved (human/artificial), SDBC adopts the *actor-role* (*role*) concept [3]. This allows for a sound and flexible modeling, where for example, a manager sending a fax would fulfill the 'secretary role'.

We have depicted the mentioned SDBC concepts in Figure 3. In positioning the concepts, we follow the classical views of Bunge [2], according to which: a (business) system is characterized by *composition* (it consists of some entities), *structure* (the entities relate to each other), and *environment* (entities and relationships outside the system). As seen from the dashed lines, we consider **role** as a composition-related concept, and **transaction** as a structure-related concept. The five communication patterns are about the transaction elaboration.

SDBC elaborates a transaction via DEMO, expresses multi-transaction structures via Petri Nets, and maps these to UML Activity diagram, in deriving a dynamic software specification model. By applying ISDL, especially in elaboration and refinement, we expect to reach a simpler and smoother representation, benefiting from ISDL's capability to model and refine a broad range of dynamic patterns [8].

In Sect. 3, we introduce an example and partially approach it through SDBC. Then ISDL is introduced and applied to the example, as a complement to SDBC (Sect. 4).

3 The FM Example

The illustrative example addressed in this section, namely the FM example, concerns the *Icomp Case*. Information about the case can be found in [13].

'FM' stands for *Financial Mediator*. The FM facilitates insurance companies. In order to use the mediator, a company should subscribe (registering for its service).

The support provided by FM to registered companies includes advice and product delivery to their customers: (i) a customer can ask FM's advice on which of the companies' products best satisfies a need; (ii) a customer can also ask FM to deliver a product, on behalf of the particular company. We focus on advice delivery only.

To receive advice from FM, the customer should firstly position his(her) request, making it clear whether it is about a health insurance, car insurance, and so on. Secondly, the customer has to specify the particular demand, for instance: to insure a car against theft with the highest possible coverage. Based on this, a *request processing unit* within the FM generates a standardized specification regarding the customer's request, which is delivered to a *match-making unit* (again within FM). The match-making unit realizes then a match, supporting in this way the FM in its advice delivery. This match is driven by a particular criterion that is chosen by the customer. For instance: a preference for the cheapest or the most reliable product available. In order to deliver such a criterion-driven match, the match-making unit uses a data bank that contains relevant rules and procedures. Besides the request processing unit's specification, the match-making unit needs as well an input from a *data search and processing unit* within FM, in order to realize the match. The data search and processing unit searches through the information that concerns registered companies,

and applies procedures to this information. This allows for a precise identification of candidate-matches, relevant to the particular customer's request. Thus, the match-making unit puts the candidate-matches list (delivered by the data search and processing unit) against the standardized request specification (delivered by the request processing unit), and realizes a match, by applying rules and procedures, as mentioned above. All presented information, concerning the current example, is partial and we only use it for illustrative purpose.

In applying SDBC, we start with the initial information structuring, identification of role types, and so on [13]. However, we omit for brevity all initial SDBC analysis and modeling steps and 'arrive' directly at a constructed structural (static) business model – Figure 4. For more information on the SDBC initial analysis and modeling, readers are referred to [13,14]. As for the mentioned model, we have constructed it, using the notations of DEMO, considering the essential concepts role and transaction.

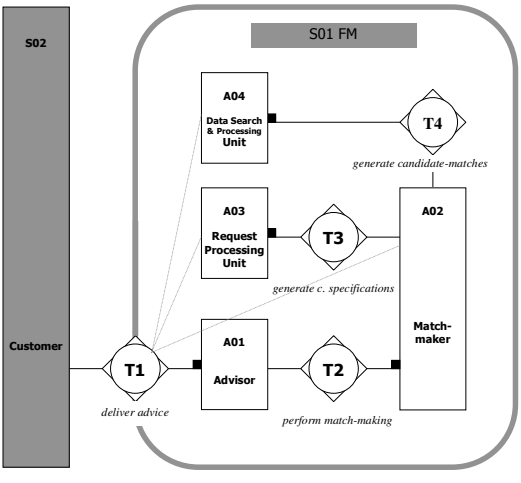


Fig. 4. Static (structural) business model in SDBC

As seen from Figure 4, an external role type is identified (Customer) as well as four internal role types (Advisor, Match-maker, Request Processing Unit, and Data Search and Processing Unit) and four transaction types (Deliver advice; Perform match-making; Generate customer's information specification; Generate candidate-matches). The rounded rectangle indicates our system's boundary. The black boxes indicate which role holds the responsibility for a transaction.

The further task is hence to proceed towards modeling that concerns the communication and dynamic viewpoints. This is to include elaboration of the modeled transactions in terms of communicative acts and coordination (staying consistent with the transaction notion – Figure 2), and also modeling of the flows of production facts.

This all is addressed in the following section which will explore the relevant value that ISDL can bring to SDBC.

4 Complementing SDBC with ISDL

The strengths of ISDL, particularly in the perspective of a SDBC-ISDL combination, are considered in this section, after a brief introduction of ISDL that is actually a language focusing on (business) process modeling at high abstraction level.

4.1 ISDL: Concepts and Notations

ISDL [6, 8] provides a small, but expressive set of basic and generic concepts for behavior modeling, aimed at modeling the behavior of systems from varying domains and at successive abstraction levels [11]. The semantics of ISDL has been defined formally; a method for conformance assessment has also been defined. Furthermore, an integrated editor and simulator is available, and tools supporting conformance assessment and model-to-model (code) transformations are being developed. Figure 5 depicts part of the behavior conceptual model of ISDL, including the *entity* concept; Figure 6 shows how these concepts are represented.

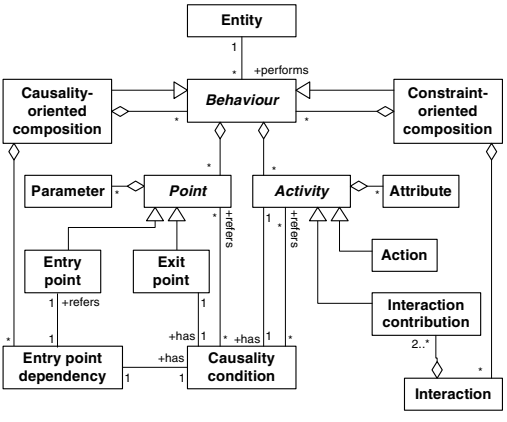


Fig. 5. ISDL concepts

The *entity* concept represents a system part that can perform some behavior. A *behavior* is essentially a set of causally related activities. An *activity* represents some unit of behavior that is atomic, i.e., cannot be split at the abstraction level at which it is defined. Further, an activity either happens, in which case reference can be made to its result, or does not happen at all, in which case no reference can be made to any result, not even to partial results. We distinguish three types of activities. An *action* is performed by a single behavior (entity). Actions are graphically expressed by ovals (or circles). An *interaction* is performed by two or more behaviors in cooperation. An interaction is expressed as two or more connected *interaction contributions* which represent the participation of the involved behaviors. Interaction contributions are expressed by oval (or circle) segments.

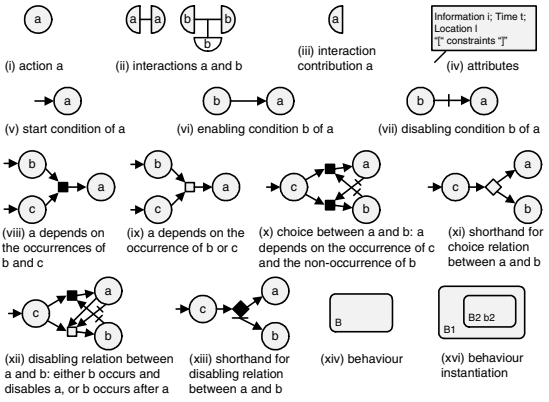


Fig. 6. ISDL language elements

An activity can have *attributes* to represent the relevant characteristics of the occurrence of the modeled real-world activity. Predefined attributes are the information, time and location attribute (see Figure 6 (iv)), representing the activity result (e.g., some information or product), the time of occurrence at which the result is available, and the location where the result is available, respectively. Constraints can be defined on the possible attribute values. The constraints also specify the relation between attribute values established in causally dependent activities. ISDL does not prescribe a language for defining attribute types and constraints, but provides bindings to existing languages that can be used for that purpose. Currently, bindings to *Z*, *Java* and *Q* exist.

Relations between activities are modeled by *causality conditions*. Each activity has a causality condition, which defines how this activity causally depends on other activities. An activity is enabled, i.e., allowed to occur, if its causality condition is satisfied. Three types of basic causality conditions are identified as illustrated in Figure 6: (v) the start condition represents that activity a is enabled from the beginning of some behavior and independent of any other activity, (vi) enabling condition b represents that activity b must have occurred before a can occur, and (vii) disabling condition ¬b represents that activity b must not have occurred before nor simultaneously with a to enable the occurrence of b. These elementary conditions can be combined using the *and*- and *or*-operator to represent more complex conditions. Figure 6 depicts also some simple examples.

Containment of one behavior by another (the composite), is represented by behavior instantiation. A behavior instantiation represents that some behavior instance is created in the context of the behavior that contains the instantiation.

4.2 Activity Refinement

An activity cannot be split at the abstraction level at which it is considered. A more detailed model of an activity can be obtained by decomposing this activity into multiple sub-activities and their relationships. The relevant characteristics of these

sub-activities can be modeled again by the activity concept (i.e., actions, interactions or interaction contributions). Therefore, the activity concept is independent of the abstraction level or granularity at which specific activities are modeled.

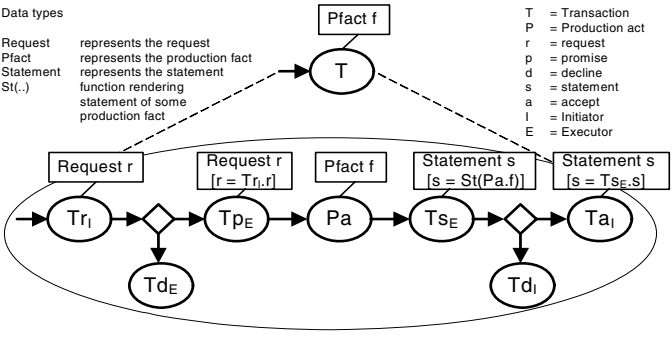


Fig. 7. The ISDL transaction models

In the context of SDBC, the activity concept is used to model transactions as well as their corresponding communication patterns. Figure 7 reflects the generic process of a transaction, modeled at two different abstraction levels. At the highest level, the transaction is represented by a single action, which models the production fact that is established. Characteristics of the production fact are modeled using action attributes. At a lower abstraction level, the transaction's communication aspects are modeled, conforming to the transaction concept (Fig. 2). Separate actions are used to model the transaction's request, promise, state, accept, and decline, and the production act. Observe that actions Td_E and Td_I correspond to the decline of a transaction followed by an unsuccessful negotiation (see Fig. 2), and actions Tp_E and Ta_I represent the promise and acceptance, respectively, which are possibly preceded by an 'initial decline' followed by a successful negotiation.

The result of the transaction behavior at the lower abstraction level should conform to the result of the transaction as modeled at the higher abstraction level. In this case, the result of the transaction behavior is either the occurrence of action Ta_I, which corresponds to the occurrence of T, or the occurrence of Td_E or Td_I, which corresponds to the non-occurrence of T. Consequently, to assess conformance one should assess whether the results as modeled by actions Ta_I and T are equivalent.

A method has been defined for ISDL to assess the conformance of any abstract behavior to a concrete behavior that refines the abstract behavior. The example in Figure 8 represents a special case of this method. For a detailed explanation of the method, interested readers are referred to [10].

4.3 Modeling the FM Example

Using the ISDL transaction models presented in Sub-section 4.2, Figure 8 depicts the modeling of the FM example (Section 3) at three successive abstraction levels. At

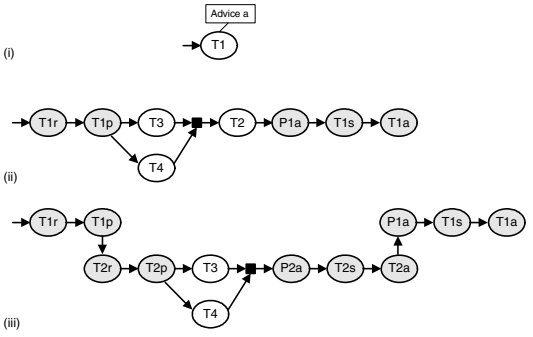


Fig. 8. ISDL models of the FM behavior

level (ii), some detail is added on how the advice is delivered, eliciting part of the internal behavior of the FM: in this case the communication aspects of T1 and the use of transactions T2, T3 and T4. More detail is added in (iii), by elaborating the communication aspects of T2. A similar elaboration can be made for T3 and T4, but has been omitted for brevity. For the same reason, action attributes are not modeled and it is assumed that transactions will not be declined. To clearly distinguish between the abstraction levels at which a transaction is modeled, the communication patterns of a transaction are indicated in grey.

The ISDL models presented so far do not consider the roles involved in each transaction. This aspect can be modeled explicitly using the *interaction* concept. For example, Figure 9 (i) models transaction T1 as an interaction between the role type Customer and FM, where roles are represented by ISDL behaviors. The interaction concept allows one to model the constraints each role may have on the possible results (production facts) of the interaction. For example, a customer may restrict the advices (s)he is interested in to car insurances, whereas FM may only consider insurances from particular companies.

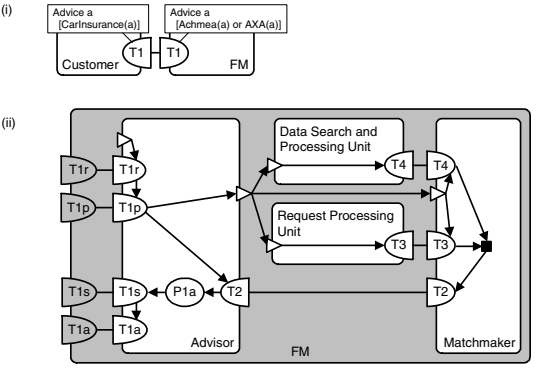


Fig. 9. ISDL models of the FM behavior

Figure 9 (ii) presents the ISDL model corresponding to the SDBC model depicted in Figure 4 (it is elicited which roles are involved in which transactions). In this case the behavior of FM is represented as a *composite behavior* (indicated in grey). Behaviors in a composite behavior can be related using: (i) *constraint-oriented composition*: interactions that relate the interaction contributions of the component behaviors; and/or (ii) *causality-oriented composition*: entry and exit points that represent a causality condition entering a behavior or a causality condition exiting a behavior, respectively. The condition that an entry point represents is associated to it via an entry point dependency. Entry and exit points are represented by triangles that point into or out of a behavior, respectively. Interaction contributions of a component behavior can contribute to interactions of their composite behavior. This is represented by drawing a line between the interaction contributions of the component and interaction contributions of the composite (having the same name in the example).

5 Analysis

As already stated, this section analyzes the suitability and adequacy of combining the SDBC approach and the ISDL language.

Our basic conclusion is that the essential value of combining SDBC and ISDL concerns the possibility to adequately grasp (driven by SDBC) real-life business aspects and realize mapping towards software specification, facilitated by a powerful language instrumentarium (ISDL) that allows one to combine (applying the powerful graphical notations of ISDL) issues concerning the communication and dynamic viewpoints; ISDL can be used at different abstraction levels and its method for conformance assessment allows one to relate successive abstraction levels. In all this, only a single formalism is needed. Further, the ISDL concepts (such as the activity concept) prove to naturally correspond to the SDBC behavior concepts (such as the transaction concept), i.e., ISDL can represent the properties modeled by SDBC concepts. Thus one can smoothly apply ISDL in the context of the SDBC approach.

Complementing SDBC by ISDL, allows not only for an adequate consideration of the notions of role and transaction – these are essential for a business process modeling driven by SDBC [15], but also for modeling transactions (through the interaction concept of ISDL) between different roles. Transactions modeled in such a way, can be defined at a high level of abstraction in contrast to e.g. modeling languages using message passing as the basic interaction concept. When using message passing, one is often forced to define transactions closer to implementation level, since one may need multiple messages to exchange the information that is required to establish the desired transaction result. Instead, the interaction concept in ISDL allows each role involved to define its constraints on the possible interaction result, while abstracting from how these constraints are implemented (e.g. through message exchange).

Therefore, this strong point of ISDL can add value in the context of SDBC – it would be possible that a transaction is decomposed into transaction contributions, defining the responsibility of each role in the transaction (still at an abstract level). When defining a transaction as an action, one abstracts from the contribution/ responsibility of each role in the transaction.

Finally, ISDL could usefully complement SDBC in a mapping towards BPEL/WSDL, for the purpose of business processes implementation using the Service-Oriented Paradigm [9,5], which is nevertheless beyond the scope of this paper.

6 Conclusions

In this paper, we have reported studies that concern the actual challenge of aligning business requirements and software functionality, driven by an adequate identification of a business model and its mapping to a software specification model. These models need to be however appropriately elicited from different perspectives. The SDBC has relevant strengths not only with respect to the business-software alignment challenge in general but also with respect to such a desired multi-viewpoint modeling. Nevertheless, SDBC appears to need further support in achieving consistency with regard to different (complementing) viewpoints, in particular: the dynamic viewpoint and its necessary elaboration (at the phase of business modeling) from the perspective of real-life communication and coordination (where issues, such as meanings, intentions, commitments, and obligations play a role). Thus, it is essential that the SDBC business modeling allows for an appropriate combination between behavior modeling and communication/coordination aspects. However, in realizing this, SDBC uses at least two modeling techniques, the transformation among which unnecessary complicates the modeling process and causes limitations. Hence, if SDBC is applied through an integrated language facility (based on one formalism and possessing powerful modeling expressiveness), the alignment between behavior models and related communication/coordination aspects would be improved.

We have identified ISDL as a good candidate in the mentioned context, given its refinement and conformance assessment capabilities as well as powerful graphical notations. In the course of the current study, we have justified this choice, by finding evidence of particular relevant strengths of ISDL. Next to that, we have demonstrated those strengths and the value of the SDBC-ISDL combination, by means of an illustrative example.

The ISDL notations, driven by one formalism, proved to work usefully in the context of the SDBC approach; they can support the approach in the alignment of behavior business models and (related) communication/coordination aspects, presenting them in a coherent whole. Further, ISDL can be used for refinement at different abstraction levels, as demonstrated in Section 4, supported by mechanisms allowing one to assess whether a refinement conforms to the original process model. Finally, with regard to service-oriented platforms, it is expected that ISDL could support SDBC in mappings to BPEL/WSDL, which although not addressed in the current work, is in the scope of further studies. Besides this, we are also planning to conduct a bigger scale real-life case study, in order to bring more practical evidence in support of our findings. Next to that, we intend to further explore the SDBC-ISDL combination, particularly from the perspective of aligning issues that concern the static and dynamic business modeling viewpoints, and to study possibilities for simulation-driven validation of business process models.

Acknowledgements

This work is part of the Freeband A-MUSE project (http://a-muse.freeband.nl). Freeband is sponsored by the Dutch government under contract BSIK 03025.

References

1. Aalst, W.V.D., Best, E.: Applications and Theory of Petri Nets. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, Springer, Heidelberg (2003)
2. Bunge, M.A.: A World of Systems, Treatise on Basic Philosophy, vol. 4. Reidel Publ. Company, Dordrecht (1979)
3. Dietz, J.L.G.: Understanding and Modeling Business Processes with DEMO. In: Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E. (eds.) ER 1999. LNCS, vol. 1728, Springer, Heidelberg (1999)
4. Dijkman, R.: Consistency in Multi-Viewpoint Architectural Design. University Press, Enschede (2006)
5. Dirgahayu, T.: Model-Driven Engineering of Web Service Compositions: a Transformation from ISDL to BPEL. University Press, Enschede (2005)
6. ISDL home: n.d, http://isdl.ctit.utwente.nl
7. Liu, K.: Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge (2000)
8. Quartel, D., Dijkman, R., Van Sinderen, M.: An Approach to Relate Business and Application Services Using ISDL. In: EDOC 2005, 9th IEEE International EDOC Enterprise Computing Conference (2005)
9. Quartel, D., Dijkman, R., Van Sinderen, M.: Methodological Support for Service-Oriented Design with ISDL. In: 2nd International Conference on Service Oriented Computing (2004)
10. Quartel, D., Ferreira Pires, L., Van Sinderen, M.: On Architectural Support for Behaviorur Refinement in Distributed Systems Design. Journal of Integrated Design and Process Science 6(1) (2002)
11. Quartel, D., Ferreira Pires, L., Van Sinderen, M., Franken, H.M.: On the Role of Basic Design Concepts in Behaviour Structuring. Computer Networks and ISDN Systems (1997)
12. Rational, UML Resource Center: http://www.rational.com
13. Shishkov, B.: Software Specification Based on Re-usable Business Components. Sieca Repro, Delft (2005)
14. Shishkov, B., Dietz, J.L.G., Liu, K.: Bridging the Language-Action Perspective and Organizational Semiotics in SDBC. In: ICEIS 2006, 8th International Conference on Enterprise Information Systems (2006)
15. Shishkov, B., Dietz, J.L.G.: Applying Component-Based UML-Driven Conceptual Modeling in SDBC. In: ICEIS 2005, 7th International Conference on Enterprise Information Systems (2005)
16. Shishkov, B., Dietz, J.L.G.: Aligning Business Process Modeling and Software Specification in a Component-Based Way, The Advantages of SDBC. In: ICEIS 2004, 6th International Conference on Enterprise Information Systems (2004)
17. Shishkov, B., Dietz, J.L.G.: Deriving Use Cases from Business Processes, The Advantages of DEMO. In: Camp, O., Filipe, J.B.L., Hammoudi, S., Piattini, M. (eds.) Enterprise Information Systems V, Kluwer Academic Publisher, Dordrecht, Boston (2004)
18. Winograd, T., Flores, F.: Understanding Computers and Cognition: A Foundation for Design. Ablex, Norwood (1986)

Semantic Resources Integration and Interoperability in the Construction Domain

Simona Barresi[1], Yacine Rezgui[1], Celson Lima[2], and Farid Meziane[1]

[1] Informatics Research Institute (IRIS), University of Salford Salford M5 4WT, UK
{S.Barresi,Y.Rezgui,F.Meziane}@salford.ac.uk
[2] Centre Scientifique et Technique du Bâtiment (CSTB), Sophia-Antipolis, France
C.Lima@cstb.fr

Abstract. Semantic resources, such as ontologies and taxonomies, are used in various domains to facilitate information exchange between people and software applications. However, in many cases, accessing a single semantic resource is no longer adequate. Consequently, new initiatives targeting the interoperability and integration of existing semantic resources have been recently increasingly investigated. This paper describes the methodology used during the FUNSIEC project to develop an Open Semantic Infrastructure for the European Construction Sector (OSIECS). This infrastructure aims to facilitate integration and interoperability between construction related semantic resources, providing a base for the development of a new generation of e-services for the domain.

Keywords: Ontology, Semantic Resources, Schema Matching.

1 Introduction

Ontologies represent an important branch of traditional philosophy concerned with "the set of things whose existence is acknowledged by a particular theory or system of thought" [1]. Gruber [2] defined an ontology as a formal, explicit specification of a shared conceptualization. An ontology therefore defines the basic terms and relations that form the vocabulary of a topic area, as well as the rules for combining these terms and relations between terms [3].

Semantic integration of heterogeneous databases [4], content-based retrieval of yellow pages as well as product catalogues [1], and management of corporate memory [5], are just some of the areas that have increasingly exploited the benefits deriving from the use of Semantic Resources (SRs)[1] and ontology engineering in general.

Specifically in the construction sector, the need for improved communication and understanding between projects' stakeholders has led to an increased development of domain specific SRs.

Construction is a knowledge intensive industry with a unique work settings and virtual organization like modus operandi [6]. This industry is heterogeneous and highly fragmented, consisting of numerous small and medium enterprises (SMEs)

[1] Semantic Resource is an expression coined in the SPICE project to refer to ontology-similar entities, such as dictionaries, taxonomies, etc.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 336–347, 2008.

working together on various building projects. One of the major consequences is the difficulty in effective and efficient communication among partners during a building project, or between clients and suppliers of construction products. Several initiatives have tried to overcome this problem by developing a variety of SRs focused on construction related terms. However, these initiatives tend to be country specific and not adapted to the multi-national nature of the sector. Also, these resources tend to be specialized for dedicated applications or engineering functions, e.g. product libraries.

In order to improve communication and information exchange between the various stakeholders during a construction project and to enable the development of a new generation of e-services for the sector, accessing a single semantic resource is no longer adequate. New initiatives, targeting the interoperability and integration of existing construction related SRs are therefore needed.

This paper describes the methodology proposed during the FUNSIEC project to develop an Open Semantic Infrastructure for the Construction Sector (OSIECS). The purpose of OSIECS is to map construction domain semantic resources between each other. FUNSIEC, funded under the European eContent programme, aimed at evaluating the feasibility of building and maintaining OSIECS. The FUNSIEC consortium consisted of CSTB, the University of Salford, and UNINOVA.

2 Methodology and Tools

As reported in the literature [7],[8], various methodologies and tools have been developed in the field of ontology. Among these, numerous methodologies have been proposed for building ontologies [9], [10], [11], [12], [13], [14], for ontology reengineering [15], for ontology learning [16], for ontology evaluation [17], [18], [19], for ontology evolution [20], [21], [22], and various methodologies and techniques for ontology mapping, merging, and alignment [23], [24].

The establishment of a consensual and unified methodology is difficult, as is suggested by the existence of a variety of methodologies, each developed for a specific purpose. Possible reasons for these difficulties could be related to the lack of maturity of the field [8] or to the problems of adapting a unique methodology to a variety of different applications, sectors and settings.

Environments supporting the development and management of ontologies through graphical interfaces have also proliferated, some providing tools for specific functions, such as integration and merging, or reasoning capabilities.

The latter category includes OntoEdit [25], OilEd [26], Protégé [27] – which support not only OIL, but also other models such as RDF_Ontolingua [28] and Ontosaurus [29]. Furthermore, as described in [24], the former category (environments supporting ontology merging) includes OntoMorph [30], Prompt [24], and Chimaera [31]. Other mapping and merging related techniques and tool reported in the literature include FCA-Merge [32], Glue and IF-Map [23].

In the context of FUNSIEC, schema matching represented a fundamental operation. In fact, a semantic infrastructure supporting integration must inevitably deal with the problems inherent to heterogeneous SRs, which often differ in both structure and terminology. Through schema matching two schemas are compared and the mapping between elements that correspond semantically to each other is produced [33], [34],

[35]. However, schema matching is considered to be a time consuming and error prone process, due to the fact that it is still predominantly performed manually. A comprehensive taxonomy, covering many of the existing approaches to automatic schema matching, is proposed in [35].

3 Semantic Resources in the Construction Sector

Among the multitude of SRs developed in the Construction sector, ranging from domain dictionaries to specialized taxonomies, some of the most notable efforts include the BS6100, bcXML, ISO 12006-3, and IFC (Industrial Foundation Classes).

The BS6100 (Glossary of Building and Civil Engineering terms), produced by the British Standards Institution, is a rich and complete glossary. It provides a comprehensive number of synonyms per term that can contribute towards any ontology development effort in the sector.

The bcXML [36] is an XML vocabulary developed by the eConstruct IST project for the Construction industry. The bcXML provides the foundation for the development of the bcBuildingDefinitions taxonomy, which can be instantiated to create catalogue contents. Through bcXML, eConstruct has enabled the creation of "requirements messages" that can be interpreted by computer applications to find suitable products and services meeting those requirements.

The ISO 12006-3 [37] defines a schema for a generic taxonomy model, which enables the definition of concepts by means of properties, to group concepts, and to define relationships between concepts.

The IFC model, developed by the IAI (International Alliance for Interoperability), has produced a specification of data structures with the aim of supporting the development of the 'Building Information Model' where all the information about the whole life cycle of a construction project would be stored and shared among the actors involved.

All of the above resources, although different in terms of formalism, scope, details and applicability, can be used in a complementary manner. Providing an infrastructure to map these resources helps to overcome problems related to SRs' different formalism and inconsistencies, and enables effective reuse of existing construction related SRs. This in turn facilitates the efficient use of knowledge within the sector and can support the implementation of advanced e-services for the construction domain.

4 FUNSIEC Methodology

Numerous methodologies for ontology mapping, merging and alignment are reported in the literature. Determining the most appropriate methodology to be applied is dependent on the nature, individual characteristics, and applications of the domain in question. In the case of an open semantic infrastructure, such as OSIECS, the applied methodology has to satisfy the following requirements: (i) Should make use of already established and recognized semantic resources. (ii) The infrastructure should be flexible and comprehensive enough to accommodate different business scenarios. (iii) The infrastructure is a living system and should allow for future expansion (including

expansion of SRs or inclusion of new SRs). (iv) The end-user perspective and evaluation should be considered when planning expansion. Consequently, building upon the strength of numerous established methodologies, a new methodology was developed during the FUNSIEC project to guide the specification of the OSIECS infrastructure.

The FUNSIEC methodology (Figure 1) comprises of the following phases: domain scoping, candidate semantic resources identification, conversion and similarity detection (OSIECS Kernel), OSIECS meta-model and model construction, testing and validation, and maintenance.

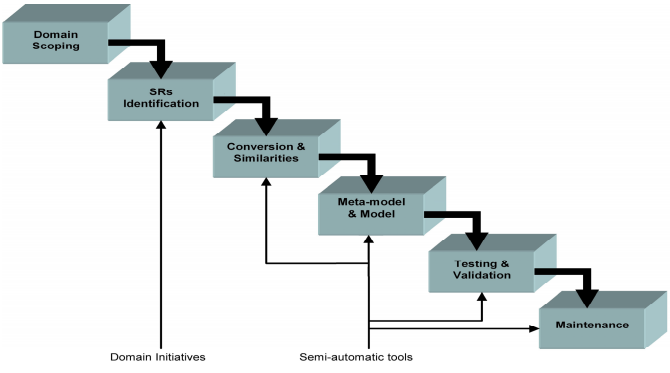


Fig. 1. FUNSIEC Methodology

The following sections provide a description of these stages. For pragmatic reasons, the conversion and similarity detection (OSIECS Kernel) and the OSIECS meta-model and model phases are here discussed as a single phase.

4.1 Domain Scoping

Scoping the domains (e.g. knowledge management, e-procurement, etc.) to be covered by OSIECS was facilitated by the use of typical scenarios that the infrastructure was expected to handle. The use of scenarios facilitates the description of the domain to be covered, how OSIECS was expected to be used, and which type of information it was expected to provide.

Two example scenarios are: (i) A designer developing a CAD drawing (IFC compliant) needs to also know the regulations to be followed in his/her project. In this case OSIECS would provide a link between the IFC tool and the e-COGNOS tool. (ii) An expert looking for information on the fire resistance of a given brick also needs to receive information on alternative products (suppliers, prices, etc.). OSIECS would then provide a link between the e-COGNOS tool and the e-Construct tool.

4.2 Semantic Resources Identification

The results of the first stage of the methodology aided the process of selecting the SRs to be included in OSIECS. Existing Construction related SRs were selected for inclusion by considering their domain and a series of other factors, such as their availability,

cost, formalism, and underlining language. The SRs included in OSIECS are the e-Cognos ontology, the IFC model, the *bcBuildingDefinition* taxonomy, and the STABU LexiCon. The latter is a vocabulary of terms for the Construction industry and as such is an implementation of ISO DIS 12006-3.

4.3 Conversion and Similarity Detection: Meta-model and Model

After selecting the SRs to be included in OSIECS, syntax related problems (data heterogeneity) were addressed by converting each SRs' meta-schemas and schemas into Web Ontology Language (OWL). This conversion facilitated the processes of semi-automatically dealing with semantic heterogeneity and detecting similarities between SRs' schemas. The conversion process produced the *"rules of conversion"* from each original formalism into OWL, which were used to create the OWL version of SRs' meta-schemas and schemas. At this stage, human intervention was required to identify the formalism used in SRs, study the semantics of the formalism, and identify syntactic elements in OWL corresponding to the syntactic elements of the formalism used in the SRs.

The next step in the construction of the OSIECS meta-model and model was to detect and validate the similarities existing among SRs' meta-schemas, and subsequently the ones existing among the different SRs' schemas. Two components were used for this purpose, the Detector of Mappings and the Validator. The Detector of Mappings used an inference engine (FONDIL[2]) to compare SRs' meta-schemas and schemas and to create lists of equivalent or subsumed concepts. The Validator component was then used to check the similarities detected. The latter was a semi-automatic process, which required the intervention of human experts to ensure that the results of the validation process were correct and to add new similarities if required.

4.4 Testing and Validation

The testing and validation phase was directed at verifying the completeness of the infrastructure in terms of the conceptualization of targeted domains, assessing the relevance of concepts and relationships, and verifying the consistency and coherency of concepts. To test and validate the OSIECS infrastructure a series of dedicated services and scenarios were implemented.

4.5 Maintenance

The final phase of the methodology is an ongoing process aimed at correcting and updating the open semantic infrastructure during its working life. Maintenance is required to eliminate errors or deficiencies in the infrastructure and to update and enrich the domains covered by OSIECS, through the integration of new SRs. In order to achieve this integration, new mappings and methods need to be considered

5 The OSIECS Triad

By the use of the FUNSIEC methodology the OSIECS *Triad* was implemented, specifically the OSIECS Kernel, the OSIECS meta-model, and the OSIECS model. This

[2] FONDIL is available at http://195.83.41.67/ondil/connect.html

section outlines the architecture of the OSIECS Kernel, a semi-automatic tool used to create both the OSIECS meta-model and model. As mentioned, the OSIECS Kernel covers two levels, the meta-schema and schema levels.

The Kernel consists of the following components: the Syntax Converter, the Semantic Analyser, the Converter, the Detector of Mappings, and the Validator. The operation of the OSIECS Kernel is depicted in figure 2. The role of the experts is to verify the results produced by the Syntactic Converter and the Semantic Analyser, as well as to help validate the lists produced by the FONDIL system.

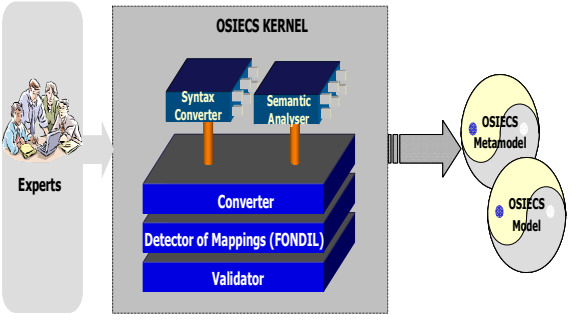


Fig. 2. The OSIECS kernel

The OSIECS meta-model and model are built by using the meta-schemas and schemas of the selected Construction related SRs. The Syntax converter and the Semantic analyser work together, using the meta-schemas and schemas of the four selected SRs as input, to produce the rules of conversion. These rules are then used by the Converter to guide the production of the OWL meta-schemas and schemas for each of the SRs in the Kernel. The Detector of Mappings is played by the FONDIL system (briefly introduced below), which works with the OWL-converted meta-schemas and schemas to produce a list of equivalent or subsumed entities. These entities are then analysed and assessed by the Validator [38].

Helped by the appropriate software tools, the experts play an essential role in the creation of OSIECS meta-model/model. They participate at both levels, taking care of: (i) the manual analysis of the SRs and their respective meta-schemas/schemas; (ii) the analysis of the *rules of conversion*; (iii) the assessment of the detection of similarities; (iv) the inspection of the validation process; and (v) the assessment of the final output.

5.1 The FONDIL System

The FONDIL system is responsible for the detection of similarities among meta-schemas/schemas within the OSIECS Kernel. In general, FONDIL provides inference services for Description Logic-based ontologies [39]. The expressiveness of such ontologies allows formalisation of the semantics of modelling languages (*e.g* UML, EXPRESS) and makes these semantics as explicit as possible. It is worth emphasising that formal and explicit semantics are crucial to automated deduction.

The FONDIL system is composed of three modules, namely *ontology management*, *mediator*, and *inference engine*. The heart of the FONDIL system is an *inference engine* that uses structural algorithms for non-standard inferences and optimised algorithms for standard inferences [39], [40]. FONDIL uses the *inference engine* to deduce new knowledge, using ontologies as the primary source of knowledge. The knowledge deduced is essentially new relations among the ontological concepts. FONDIL initially considers that the *ontology manager* needs some help to exploit all the possible relationships among the concepts within a single ontology. This help is even more necessary when considering several SRs that were developed independently from each other. The relationships among them (if they exist) are usually implicitly defined. These relationships can be viewed more as knowledge to be detected rather than knowledge to be predefined in the SRs. FONDIL's role within OSIECS is to assist in the refinement of the semantic mappings detected among the SRs [40].

5.2 Syntactic Conversion and Semantic Analysis

The mapping process involves three main aspects of SRs: the structures, the syntax, and the semantics. Before describing the process of creation of the OSIECS meta-model and model, it is worth noting that the meta-schemas used to form OSIECS are originally represented in different formalisms. EXPRESS is used in ISO 12006-3 and IFC, and UML is used in e-COGNOS and bcXML.

To solve syntax problems, the recommended solution is to represent the original SRs in neutral format; this can be achieved through conversion if necessary. The converted versions are then free of syntactical problems. Structural and semantic-related problems are solved through a semi-automatic process [41].

As depicted in figure 2, the *Converter* works with the meta-schemas/schemas in their original formats and produces the corresponding OWL versions. The experts play a strategic role in this phase, since they analyse the SRs' meta-schemas/schemas and create a set of conversion rules (in Java) that allows conversion of each entity from their original format into OWL. This transformation must preserve the semantics of the converted entities. The set of rules is then used by the JavaCC[3] tool, which generates "transformers" capable of automatically translating any meta-schema/schemas written in the original format into OWL. During the OSIECS development, two "transformers" were generated to support the translations of both EXPRESS and UML to OWL.

5.3 Detection of Mappings

As mentioned, the *Detector of mappings* uses the FONDIL inference engine to detect the similarities between each pair of concepts belonging to two different SRs. The similarity between two concepts is defined in four levels, according to its granularity.

Let us consider two concepts *C1* and *C2* belonging to two meta-schemas. Firstly, the inference engine verifies if they are equivalent according to the OWL semantics. In case of equivalence, this result will be sent to the *Validator*. Otherwise, these concepts are sent to the *Subsumption Detection* component that will check if one concept is subsumed by the other. If the concepts are not subsumed to each other, the similarity

[3] JavaCC is available at https://javacc.dev.java.net

between them is evaluated by the *Intersection Detection*, *LCS and Difference Detection* components, which will deal with intersections, unions and differences among the concepts. This allows a more accurate detection of similarities between the two concepts. The similarities between the meta-schemas must be then validated in order to produce the OSIECS meta-model.

5.4 Matching the Entities

The similarities found in the previous stage are used over the schemas of the SRs, following a specialisation process. For instance, let *A* be an entity from the e-COGNOS meta-schema and *B* an entity from the ISO 12006-3 meta-schema. Thus, *S(A, B)* represents a similarity between these entities. This similarity is then matched to the entities of the correspondent e-COGNOS and LexiCon schemas, *S'(a, b)*. All the entities matched at the schema level of the selected SR compose the OSIECS model.

5.5 OSIECS Meta-model and Model

Basically, the OSIECS meta-model and model are mapping tables that identify and establish the semantic correspondence between the entities forming the SRs. The OSIECS meta-model is the set of tables mapping the meta-schemas of the SRs forming OSIECS, while the OSIECS model is the set of tables mapping the schemas of SRs forming OSIECS.

The OSIECS meta-model and model support two different areas of application. The meta-model supports the creator of SRs who intends to map several SRs to create a new one. The model supports the use of SRs developed by different providers (in different formats) which need to be used together in a transparent way for the user (e.g. a query written in one specific format is, indeed, going to target several SRs simultaneously thanks to the mappings established by the model).

The OSIECS meta-model and model were both created during the FUNSIEC project and evaluated by the experts involved in the creation process. According to the description logic-based system, if concept names match, the respective concepts are considered to be equivalents. Considering for example the bcXML and eCognos meta-schemas, some concept names are identical (e.g. 'Object'), and therefore match. However, these 'Object' concepts have different meanings. To overcome this issue and to distinguish each concept name, different namespaces were added to each ontology (or a prefix to each concept).

The results achieved from the OSIECS meta-model and model showed that subsumption mappings were more numerous than equivalence mappings. The reason for this lies in the differences between the SRs used. A predominance of equivalent concepts, would have been expected if the SRs used were very similar, however, this was not the case, even if the SRs used represent knowledge from the same domain. More information on the results derived from the assessment of the OSIECS kernel, together with examples of the mapping table obtained, can be found in [44].

6 e-Service Provision

Although OSIECS primary purpose is to map construction related SRs, it also facilitates the provision of construction related e-services. Current software application-based

collaboration requires integration through shared semantic resources. FUNSIEC argues that attention needs to be paid to direct support for business transactions and processes. This would enable organizations to migrate their legacy/commercial application systems, articulated around proprietary semantic resources, to higher order interoperable applications supporting real business processes [42].

The OSIECS Kernel can provide e-services over SRs for the construction sector. The initial list of OSIECS e-services concentrates on the enhancement of the OSIECS meta-model and model, including the following services: (i) Automatic conversion of SRs written in EXPRESS/UML (both meta-schemas and schemas) to OWL. (ii) Verification of the '*compatibility level*' of a given SR (represented in EXPRESS/UML) regarding the OSIECS meta-model/model. (iii) Mapping report between SRs represented in OWL [43]. Considering that OSIECS can be promoted and adopted, at least at the experimentation level, future e-services are to consider the creation and publication of e-catalogues, the creation and management of SRs (taxonomies and ontologies, in this case), and semantic mapping amongst SRs.

7 Conclusion

This paper presented part of the research carried out in the FUNSIEC project. By using the FUNSIEC methodology experts successfully developed the OSIECS Kernel, mapped four SRs between each other, and created the OSIECS meta-model and the OSIECS model. The FUNSIEC methodology is expected to be used in the future to integrate new SRs into the existing pool of resources already included in OSIECS. This will require new mappings and methods to be considered.

The FUNSIEC approach began with the characterisation of the domains, selection of pertinent SRs and the subsequent analysis of their meta-schemas/schemas. The SRs selected to form OSIECS, were converted into OWL. This process was performed semi-automatically by experts that extracted a set of conversion rules to feed the JavaCC, which created the respective *transformers* that were consequently used to perform the conversion of the SRs into OWL. The converted meta-schemas were semantically compared and mapped using the FONDIL system. The final output was the OSIECS meta-model. The OSIECS model was then produced using the same process.

Finally, another important aspect of the OSIECS Kernel is its ability to support a multitude of e-services. For the time being, the OSIECS Kernel provides e-services targeting the enrichment of the OSIECS meta-model/model. More services are expected to be included in this list in the near future.

Acknowledgements

The authors would like to thank the members of the FUNSIEC consortium for their valuable contributions to this research, and the European Commission for the financial support under the IST and eContent programs.

References

1. Guarino, N., Masolo, C., Vetere, G.: OntoSeek: ContentBased Access to the Web. IEEE Intelligent Systems 14(3) (May/June 1999)
2. Gruber, T.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies (IJHCS) 43(5/6), 907–928 (1994)
3. Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T., Swartout, W.R.: Enabling Technology for Knowledge Sharing. AI Magazine 12(3), 36–56 (1991)
4. Partridge, C.: The Role of Ontology in Integrating Semantically Heterogeneous Databases, Technical Report 05/02, LADSEB-CNR, Padova, Italy (June 2002)
5. Corporate Memory Management through Agents (CoMMA). In: Proceedings of E-Work & E-Business, Madrid (2000)
6. Rezgui, Y.: Review of Information and Knowledge Management Practices - State of the Art in the Construction Industry. The Knowledge Engineering Review Journal 16(3), 241–254 (2001)
7. Corcho, O., Fernando-Lopez, M., Gomez-Perez, A.: Methodologies, Tools and Languages for Building Ontologies. Where is their Meeting Point? Data and Knowledge Engineering 46, 41–64 (2003)
8. Fernandez-Lopez, M.: Overview of Methodologies for Building Ontologies. In: Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm (1999)
9. Aussenac-Gilles, N., Biébow, B., Szulman, S.: Revisiting Ontology Design: a Methodology Based on Corpus Analysis. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 396–408. Springer, Heidelberg (2000)
10. Blazquez, M., Fernandez, M., Garcia-Pinar, J.M., Gomez-Perez, A.: Building Ontologies at the knowledge level using the ontology design environment. In: Proceeding of Knowledge Acquisition Workshop, Banff, Canada (1998)
11. Holsapple, C.W., Joshi, K.D.: A Collaborative Approach to Ontology Design. Communication of the ACM 45(2), 42–47 (2002)
12. Kayed, A., Colomb, R.M.: Extracting Ontological Concepts for Tendering Conceptual Structures. Data and Knowledge Engineering 40, 71–89 (2002)
13. Pinto, H.S., Gomez-Perez, A., Martins, J.P.: Some Issues on Ontology Integration. In: Proceedings of the IJCAI-99 Workshop on Ontologies and Problem Solving Methods (KRR5), Stockholm, Sweden (1999)
14. Pinto, H.S., Martins, J.P.: Reusing Ontologies. In: Proceedings of AAAI 2000 Spring Symposium Series, Workshop on Bringing Knowledge to Business Processes, SS-00-03, pp. 77–84. AAAI Press, Stanford (2000)
15. Klein, M.: Combining and Relating Ontologies: An Analysis of Problems and Solutions. In: Gomez-Perez, A., Gruninger, M., Stuckenschmidt, H., Uschold, M. (eds.) IJCAI 2001. Workshop on Ontologies and Information Sharing, Seattle, USA (2001)
16. Kietz, J.U., Maedche, A., Volz, R.: A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. In: EKAW 2000 Workshop on Ontologies and Texts, Juan-Les-Pins: CEUR Proceedings, vol. 51 (2000)
17. Gruninger, M., Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal (1995)
18. Guarino, N., Welty, C.: Ontological Analysis of Taxonomic Relationships. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) ER 2000. LNCS, vol. 1920, pp. 210–224. Springer, Heidelberg (2000)

19. Kalfoglou, Y., Robertson, D.: Use of Formal Ontologies to Support Error Checking in Specifications. In: Fensel, D., Studer, R. (eds.) EKAW 1999. LNCS (LNAI), vol. 1621, pp. 207–224. Springer, Heidelberg (1999)
20. Klein, M., Fensel, D.: Ontology Versioning on the Semantic Web. In: SWWS 2001. First International Semantic Web Workshop, Stanford (2001)
21. Klein, M., Fensel, D., Kiryakov, A., Ognyanov, D.: Ontology Versioning and Change Detection on the Web. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, Springer, Heidelberg (2002)
22. Noy, N.F., Klein, M.: Ontology Evolution: Not the Same as Schema Evolution. Technical Report SMI-2002-0926, Stanford (2002)
23. Kalfoglou, Y., Schorelmmer, M.: Ontology Mapping: the State of the Art. The Knowledge Engineering Review 18(1), 1–32 (2003)
24. Noy, N.F., Musen, M.A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: AAAI 2000. 17th National Conference on Artificial Intelligence, Austin (2000)
25. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 221–235. Springer, Heidelberg (2002)
26. Bechhofer, S., Horrocks, I., Goble, C., Stevens, R.: OilEd: A Reasonable Ontology Editor for the Semantic Web. In: Baader, F., Brewka, G., Eiter, T. (eds.) KI 2001. LNCS (LNAI), vol. 2174, pp. 396–408. Springer, Heidelberg (2001)
27. Grosso, E., Eriksson, H., Fergerson, R.W., Tu, S.W., Musen, M.M.: Knowledge Modelling at the Millennium - the Design and Evolution of Protégé-2000. In: KAW 1999. Proc. the 12th International Workshop on Knowledge Acquisition, Modeling and Mangement, Banff (1999)
28. Farquhar, A., Fikes, R., Rice, J.: The Ontolingua Server: A Tool for Collaborative Ontology Construction. In: KAW 1996. Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Ban, pp. 44.1–44.19 (1996)
29. Lenat, D.B., Guha, R.V.: Building Large Knowledge-Based Systems: Representation and Inference. In: The Cyc Project, Addison-Wesley, Boston (1990)
30. Chalupsky, H.: OntoMorph: a Translation System for Symbolic Knowledge. In: KR 2000. Seventh International Conference on Principles of Knowledge Representation and Reasoning, pp. 471–482. Morgan Kaufmann, San Francisco (2000)
31. McGuiness, D.L., Fikes, R., Rice, J., Wilder, S.: An Environment for Merging and Testing Large Ontologies. In: Seventh International Conference on Principles of Knowledge Representation and Reasoning 5KR 2000, Breckenridge, USA (April 12-15, 2000)
32. Stumme, G., Ad Maedche, A.: Ontology Merging for Federated Ontologies on the Semantic Web. In: Gomez-Perez, A., Gruninger, M., Stuckenschmidt, H., Uschold, M. (eds.) IJCAI 2001. Workshop on Ontologies and Information Sharing, Seattle, USA (August 4-5, 2001)
33. Li, W., Clifton, C.: Semantic Integration in Heterogeneous Databases Using Neural Network. In: Proceedings of the 20th International Conference On Very Large Data Bases, pp. 1–12. Morgan Kaufmann Publishers, San Francisco (1994)
34. Milo, T., Zohar, S.: Using Schema Matching to Simplify Heterogeneous Data Transaction. In: Proceedings of the 24th International Conference On Very Large Data Bases, pp. 122–133. Morgan Kaufmann Publishers, San Francisco (1998)
35. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal 10(4) (December 2001)

36. eConstruct, Final Edition of the bcXML Specifications (2001), http://www.econstruct.org/6public/bcxml_cd/publicdeliverables/d103_v2.pdf
37. ISO DIS 12006-3. Organisation of Information About Construction Works. Part 3. Framework for object-oriented information. International Standards Organisation (2004)
38. Lima, C., Storer, G., Zarli, A., Ferreira-da-Silva, C.: Towards a Framework for Managing Standards-Based Semantic E-Resources in the European Construction Industry. In: CRC. Proceedings of Construction Research Congress, San Diego, US (April 2005)
39. Le Duc, C.: Transformation d'Ontologies Basées sur la Logique de Description - Application dans le Commerce Electronique. Thesis (PhD), Université de Nice (2004)
40. Lima, C., Ferreira da Silva, C., Sousa, P., Pimentão, J.P., Le-Duc, C.: Interoperability among Semantic Resources in Construction: Is it Feasible? In: CIB-W 1978 Conference, Dresden, Germany (July 2005)
41. Lima, C., Ferreira-da-Silva, C., Le Duc, C., Zarli, A.: A Framework to Support Interoperability among Semantic Resources. In: Proceedings of INTEROP-ESA 2005, Geneva, Switzerland (February 2005)
42. Rezgui, Y., Meziane, F.: A Web Service Implementation of a User Centred Knowledge Management Platform. In: International Journal of Intelligent Information Technologies, vol. 1(4), Idea Group Publishing, USA (2005)
43. Barresi, S., Rezgui, Y., Lima, C., Meziane, F.: Architecture to Support Semantic Resources Interoperability. In: Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems, pp. 79–82. ACM Press, New York (2005)
44. D2.2 - The OSIECS Infrastructure - Assessment and Validation. Feasibility Study for a Unified Semantic Infrastructure in the European Construction Sector, http://liris.cnrs.fr/publis/?id=2188

Part IV

Software Agents and Internet Computing

Establishing the Semantic Web Reasoning Infrastructure on Description Logic Inference Engines

Dimitrios A. Koutsomitropoulos, Dimitrios P. Meidanis, Anastasia N. Kandili, and Theodore S. Papatheodorou

High Performance Information Systems Laboratory, School of Engineering
University of Patras, Buidling B, 26500 Patras-Rio, Greece
{kotsomit,dpm,nkandylh,tsp}@hpclab.ceid.upatras.gr

Abstract. The recent advent of the Semantic Web has given rise to the need for efficient and sound methods that would provide reasoning support over the knowledge scattered on the Internet. Description Logics and DL-based inference engines in particular play a significant role towards this goal, as they seem to have overlapping expressivity with the Semantic Web de facto language, OWL. In this paper we argue that DLs currently constitute one of the most tempting available formalisms to support reasoning with OWL. Further, we present and survey a number of DL based systems that could be used for this task. Around one of them (Racer) we build our Knowledge Discovery Interface, a web application that can be used to pose intelligent queries to Semantic Web documents in an intuitive manner. As a proof of concept, we then apply the KDI on the CIDOC-CRM reference ontology and discuss our results.

1 Introduction

Regarding the success of the Semantic Web, it may be encouraging that relevant applications and systems that utilize its standardized "toolkit" of languages and specifications tend to proliferate day by day. Even these very specifications are subject to ongoing research that attempts to push the limits of the current Semantic Web idea some steps further. Nevertheless, measuring the success of the Semantic Web could also be regarded by the point of view of the goals achieved so far: Web knowledge management; semantic resource description; and distributed knowledge discovery, as one of the most prominent. In order for this promise not to be failed, Semantic Web surely could only benefit from efficient and sound methods that would provide reasoning support for its underlying knowledge.

Description Logics and DL-based inference engines in particular play a significant role towards this goal, as they seem to have overlapping expressivity with the Semantic Web de facto language, OWL. In addition, implemented algorithms and reasoning systems for DLs already exist that could be used to provide knowledge discovery facilities on the Semantic Web. Combined, these two facts make the use of DLs one of the most tempting available formalisms to typically support reasoning with OWL.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 351–362, 2008.

In this paper we first compare DL-based systems with alternatives based on other formalisms, like rule based systems and theorem provers, and argue that DLs are currently the most suitable means to build reasoning services for the Semantic Web. Then, we present and survey five popular systems from the DLs world and evaluate them in terms of their availability, expressivity and ability to reason with individuals (ABox support): Cerebra, FaCT, FaCT++, RACER and Pellet. In order to demonstrate the ability to perform Semantic Web reasoning using DL based systems, we have chosen one of the inference engines above as the core of our Knowledge Discovery Interface (KDI). The KDI is a web-distributed application that can be used to pose intelligent queries to Semantic Web documents in an intuitive manner. In order to answer these queries, the KDI relies on the reasoning services provided by the underlying inference engine. Finally we construct and use some instances of the CIDOC-CRM ontology, which we then feed in to the KDI and discuss the results from a series of intelligent queries posed.

The rest of this paper is organized as follows: In section 2 we compare available reasoning formalisms against DLs and report corresponding systems. Then, in section 3, we survey a number of DL-based systems and explain our evaluation criteria. The KDI is presented in section 4; we describe its functionality and architecture followed by some experimental results on the CIDOC-CRM ontology that demonstrate its capabilities. Finally, section 5 summarizes the conclusions from our work.

2 Inference Systems for the Semantic Web

As OWL does not natively support or suggest a reasoning mechanism, we have to rely on an underlying logical formalism and a corresponding inference engine. In this section we first briefly compare some inference methods for the Semantic Web, alternative to DLs, with DL-based systems. We present the formal relation of DLs with OWL and then discuss a number of systems based either on full First Order Logic (FOL) or rule-based systems.

Description Logics (DLs) form a well defined subset of First Order Logic (FOL). OWL Lite and OWL DL are in fact very expressive description logics, using RDF syntax. Therefore, the semantics of OWL, as well as the decidability and complexity of basic inference problems in it, can be determined by existing research on DLs. OWL Full is even more tightly connected to RDF, but its typical attributes are less comprehensible, and the basic inference problems are harder to compute (because OWL Full is undecidable). Inevitably, only the examination of the relation between OWL Lite/DL with DLs may lead to useful conclusions. On the other hand, even the limited versions of OWL differ from DLs, in certain points, including the use of namespaces and the ability to import other ontologies.

It has been shown [6] that OWL DL can be reduced in polynomial time into the description logic *SHOIN*(D), while there exists an incomplete translation of *SHOIN*(D) to *SHIN*(D). This translation can be used to develop a partial, though

powerful reasoning system for OWL DL. A similar procedure is followed for the reduction of OWL Lite to *SHIF*(D), which is completed in polynomial time as well. In that manner, inference engines like FaCT and RACER can be used to provide reasoning services for OWL Lite/DL.

The selection of a DL system to conduct knowledge discovery is not the only option. A fairly used alternative are inference systems that achieve reasoning using applications based in **FOL (theorem provers)**. Such systems are Hoolet, using the Vampire theorem prover, Surnia, using the OTTER theorem prover and JTP [2] used by the Wine Agent. Inference takes place using axioms reflecting the semantics of statements in OWL ontologies. Unfortunately, these axioms often need to be inserted manually. This procedure is particularly difficult not only because the modeling axioms are hard to conceive, but also because of their need for thorough verification. In fact, there are cases where axiom construction depends on the specific contents of the ontology [9].

Another alternative is given by **rule based reasoning systems**. Such systems include DAMLJessKB [10] and OWLLisaKB. The first one uses Jess rule system to conduct inference on DAML ontologies, whereas the second one uses the Lisa rule system to conduct inference on OWL ontologies. As in the case of theorem provers, rule based systems demand manual composition of rules that reflect the semantics of statements in OWL ontologies. This can also be a possible reason why such systems can presently support inference only up to OWL Lite.

On the other hand, neither the currently available Description Logic systems nor the algorithms they implement, support the full expressiveness of OWL DL. Even if such algorithms are implemented, their efficiency will be doubtful, since the corresponding problems are optimally solved in non-deterministic exponential time. In [7] a decision procedure is presented for the *SHOIQ* description logic; this algorithm is claimed to exhibit controllable efficiency and is currently implemented in two high-end inference engines (Pellet and FaCT++).

DLs seem to constitute the most appropriate available formalism for ontologies expressed in DAML+OIL or OWL. This fact also derives from the designing process of these languages. In fact, the largest decidable subset of OWL, OWL DL, was explicitly intended to show well studied computational characteristics and feature inference capabilities similar to those of DLs. Furthermore, existing DL inference engines seem to be powerful enough to carry out the inferences we need.

3 DL Systems Evaluation

Having discussed the pros and cons of DLs as the underlying reasoning formalism for the Semantic Web we will now examine five inference engines based on DLs that could be used to provide reasoning services in OWL ontologies: Cerebra, FaCT, FaCT++, RACER and Pellet. Our evaluation, summarized in Table 1, is carried out in terms of their availability, expressiveness, support for OWL, reasoning with instances (ABox) and interconnection capabilities provided.

Table 1. Comparison summary of some DL-based inference engines

	Availability	Connectivity	Reasoning Strength	Native OWL support (syntax)	Reasoning with instances (ABox)
Cerebra	Commercial	RMI, SOAP	*SHIQ*	Yes	No
FaCT	Free	CORBA, DIG/1.0	*SHIQ*	No	No
FaCT++	Free	JNI, DIG/1.1	*SROIQ*(D)	No	Yes
Racer	Free (before 1.8)	TCP, DIG/1.1	*SHIQ*(D)	Yes	Yes
Pellet	Free	DIG/1.1	*SROIQ*(D)	Yes	Yes

3.1 Cerebra

Cerebra, by Cerebra Inc. (formerly Network Inference, now ingested by webMethods and absorbed in its Fabric product) is a commercial system, providing reasoning as well as ontology management features. Cerebra differs from traditional DL based systems, in that it provides some extra features that may be desirable in a production environment. Nevertheless, its expressive power is by no means exceedingly different.

Indeed, one interesting feature of Cerebra is the ability to add persistency to the knowledge bases that is able to process. Cerebra can load OWL documents either from the local file system or directly from the Web, provided the corresponding URL. The ontology information is stored, following an internal data model, in a relational database and can then be reloaded if needed.

Cerebra provides for connecting with client applications written in Java or .NET. Further, any web service may use its functionality through its SOAP interface. Clients written in Java can connect to the system either through RMI or SOAP, by using the classes provided by Cerebra for this purpose. For .NET, Cerebra provides a .dll library which can be used to connect with the SOAP interface. In both cases there is an API that provides for processing, managing and posing queries to ontologies. Query composition, especially when involving instances, follows to some extent the XQuery standard.

To our knowledge, there is no formal documentation for Cerebra's expressive power. It is known however that Cerebra's internal semantic model for conducting inferences is based on DLs. Our experimental evaluation of the system has shown that, for the taxonomic part of the ontology (the TBox), Cerebra supports nearly all constructors and axioms for classes and roles (including set-theoretic operations) that would normally classify it to OWL DL expressiveness level. However, further experimentation with the system has revealed the following:

- Symmetric roles cannot be recognized. This was confirmed as a system's bug.
- Minimum cardinality greater than 1 cannot be expressed (e.g. minCardinality=2), which is especially useful when modelling number restrictions.

- The most important, inference based on instances (ABox) is not supported. One possible exception is the instanceProperty function. However, given a class and a role, instanceProperty returns all the instance pairs that are inferred to be related through the given role, and its left argument comes from the given class.

The above rank Cerebra's expressiveness at *SHIQ* level, at most. On the other hand, the relational model used by Cerebra allows the submission of very powerful instance-retrieval queries, based on XQuery syntax. These queries may involve data types as well, like strings and numbers, as well as operands between them (equation, comparison). Still, the results are based only on the explicitly expressed information of the ontology, and not on information that could be inferred.

3.2 FaCT

FaCT [8] is a freely available reasoning software, that is being developed at Manchester University under Prof. Ian Horrocks. Initially, FaCT supported the *SHF* description logic and then evolved to include *SHIF* and finally *SHIQ*. FaCT's latest versions allow its interconnection with other applications following the client – server model through a CORBA interface. Furthermore, they support the DIG/1.0 standard, which prescribes a simple communication protocol through the exchange of XML requests and responses over HTTP.

FaCT implements optimized complete and sound algorithms to solve the subsumption problem in the description logics mentioned. Even though a pioneering system in its age, whose performance used to outrank other traditional DL systems, FaCT's lack of support for inference in the ABox renders it inappropriate for OWL. Indeed, during our evaluation we attempted to convert a simple OWL ontology to the intermediate form supported by FaCT. This conversion has been achieved using a tool available through the WonderWeb IST project, under which the next version of the system (FaCT++) is also developed. This conversion had the following results:

- Individuals are transformed into primitive concepts.
- Relations between individuals are not preserved.
- The new concepts that were created to represent individuals are now subsumed by the concepts the individuals initially belonged to.

Besides the lack of support for ABox and data types (concrete domains), the system is also syntactically incompatible with OWL. Apart from the intermediate, lisp-like knowledge base format, FaCT also supports ontologies in XML format, following a proprietary schema. Naturally however, the transition to and from OWL would result in significant information loss.

3.3 FaCT++

Many of FaCT's disadvantages are being coped with in the system's next version, FaCT++ [14], which has been developed as a part of the WonderWeb project. FaCT++ differs from FaCT in many aspects. It is a re-implementation of FaCT in C++, featuring however greater expressivity, aiming ulimately to support OWL DL.

FaCT++ currently implements the decision procedure for *SHOIQ* [7] and also the proposed rule-set extensions for OWL 1.1. These, along with its new datatype reasoning architecture, match FaCT++ expressiveness with the *SROIQ*(D) logic.

FaCT++ earlier versions (before 1.0) do not natively support XML/OWL syntax; however a transformation tool to the Lisp intermediate form supported by FaCT++ is provided. Individuals (and thus nominals) survive this transformation, but they are not yet fully supported, as they are all approximated as primitive concepts.

Recent versions appear to mimic OWL abstract syntax through an ASK/TELL manner, but full OWL syntax support is left as an implementation goal [14]. In addition the reasoner binaries do not seem to operate in batch mode or offer command-line loading features. Running the FaCT++ binary initiates the DIG server on a user-specified port.

As a C++ program, in order to support the OWL API (Jena is not yet supported), Fact++ has been extended with a Java Native Interface (JNI) [5]. Still this requires direct in-memory deployment of the reasoner.

3.4 RACER

RACER [3] is an inference engine for very expressive DLs. It is the first system in its category to support reasoning in ABox as well as TBox, and this used to be its main asset in comparison to the other inference engines.

RACER is being developed by profs. Volker Haarslev and Ralf Moeller in Concordia University and Hamburg Technical University respectively. It is freely available for research purposes, while a corporation has been established for the commercial exploitation of the system (Racer v1.8+ aka RacerPro).

RACER's communication with other applications is achieved through the TCP/IP interface provided or through HTTP, since the system supports the DIG/1.1 standard. For TCP communication there are APIs available in C++ as well as in Java. In addition, RACER can been run in "file mode", where the ontology and queries files are given as parameters from the command line.

Apart from the lisp-like knowledge base format, RACER can load and process natively ontologies written in XML, RDF, RDFS, DAML+OIL and finally OWL (since version 1.7.7). The underlying description logic is *SHIQ*(D), including instances (ABox). In fact, RACER expressiveness is superior to OWL DL in regard to qualified number restrictions and concrete domains.

Indeed, RACER implements algorithms for conducting inferences based on min/max relations between integers, linear polynomial equalities and inequalities of reals, non-linear polynomial equations of complex numbers and string comparison. On the other hand, OWL allows only expressing equality between an individual and an instance of the concrete domain.

However, OWL semantics are more expressive than the RACER language as far as *nominals* are concerned, because they are not supported by the system. This seems to be the main problem that prevents full compatibility with OWL DL. RACER deals with nominals by creating a new concept for each of them and making the corresponding individual an instance of this new concept.

3.5 Pellet

Pellet [13] is a tableaux-based DL reasoner written in Java. Pellet started as a research project of MindSwap Lab at UMBC and has been the first system that implemented the decision procedure for the *SHOIQ* description logic, thus supporting OWL DL as a whole. Pellet is now open-source and is maintained by Clarck & Parsia LLC.

The reasoner may operate as a command-line standalone application and can directly load and handle OWL documents. As a Java application, it also offers and API that can be used along with other interfaces such as the Jena Toolkit and the OWL API (WonderWeb). As a distributed module, Pellet offers only a DIG/1.1 interface which has limited OWL support. Pellet libraries are also bundled with the SWOOP editor and the alpha-stage Protégé 4.0.

Pellet exhibits a sound and complete reasoning behavior with OWL DL, meaning that it can also handle nominals successfully. Also, after version 1.4, support for the rule extensions proposed with OWL 1.1 has been added. Reasoning with XML Schema datatypes is also offered, therefore *SROIQ*(D) is the underlying logic supported.

3.6 Discussion

FaCT++ and Pellet are currently the only two DL-based engines that appear to fully support the decidable subset of OWL. However they only support DIG 1.1, which is insufficient for full OWL DL support [1], a fact that mostly drives the upcoming 2.0 specification.

DIG 1.1 communication takes place over HTTP and there is no other TCP/IP-like connectivity support; rather, for these reasoners to be utilized by a tool or application, one may use a programmatic API (e.g. Jena or the Manchester API) that interfaces these reasoners as direct in-memory implementations [4]. This approach may have the advantage of reducing the message-passing load of the DIG protocol, but surely is insufficient for developing truly decentralized Web applications and services for the Semantic Web. As DIG 2.0 specification that would solve the aforementioned problems is currently in flux, these reasoners cannot be used in developing a distributed web service for Semantic Web Knowledge Discovery that would fully support OWL DL.

For the purposes of our work, we have opted for RACER as a DL-based reasoning back-end. RACER used to be dominant in terms of expressivity and interface abilities among DL-reasoners, when Pellet was not even existent. Now, RACER, being freely available for non-commercial purposes, is the only free engine, with expressive strength closest to OWL DL that exposes/maintains an independent, full-featured, IP-compatible communication interface. Indeed, its utilization in the KDI produced a number of interesting results, some of which are presented in subsection 4.2.

4 DL-Based Knowledge Discovery

In this section we demonstrate the use of DLs for knowledge discovery on the Semantic Web. First we give a general description of the KDI and the main technologies that were used, along with a brief description of its functionality. Then,

using the KDI, we present two experimental inferences on CIDOC-CRM instances expressed in OWL DL, and their results.

4.1 The Knowledge Discovery Interface

The KDI is a web application, providing intelligent query submission services on Web ontology documents. We use the word *Interface* in order to emphasize the fact that the user is offered a simple and intuitive way to compose and submit queries. In addition, the KDI interacts with RACER to conduct inferences. The interface design follows the traditional 3-tier model, with an important variation: Where a database server would be typically used, we now use a knowledge base management system (Figure 1). Note that each of the three levels may be physically located on different computer systems.

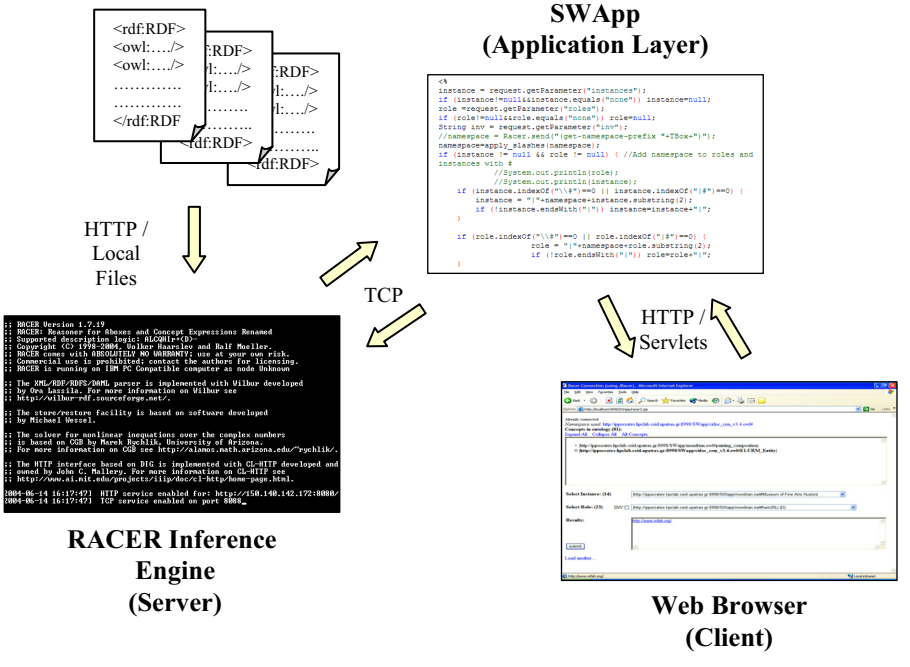


Fig. 1. The three tiers of the Knowledge Discovery Interface

The interface can load OWL documents that are available either on the local file system, or on the Internet. A temporary copy of every document is stored locally on the application server and is then loaded by the **knowledge base server** (RACER). RACER creates and stores in memory an internal model for each ontology that it classifies. Classification takes place once for each ontology, during its initial loading. Furthermore, other documents imported by the ontology may be loaded too.

The Interface business logic was implemented using the Java programming language, as well as JSP, JavaBeans and Java Servlets technologies. Tomcat (version

5.0) was used as an **application server**. Business logic is mostly responsible for document loading, proper rendering of the ontological information to the user, composition and submission of queries and formulation of the results. Ontological data and reasoning results are fetched by interacting with RACER over the TCP/IP protocol. This interaction is greatly facilitated through the JRacer API. The latter has been modified in places, mainly in regard to the processing of web documents links and to the processing of synonym concepts.

The user interacts with the **client front-end**, where the appropriate JSP pages are rendered by the browser. Communication with the application layer is conducted over the HTTP protocol, using forms. At the same time, servlets are used for the administration of multiple user requests and for controlling simultaneous access. Furthermore, when a loaded ontology is not used any more, it is erased from memory, in order to improve the utilization of system resources. For a further description of the KDI, the reader is referred to [11].

4.2 Results

In the following we present the results from two different inference actions performed using the KDI, so as to demonstrate its capabilities as well as its limitations. In order to conduct these inferences we use the CIDOC Conceptual Reference Model as our knowledge base. This approach is detailed in [12].

Firstly, we ported version 3.4 of the CRM to OWL format. Secondly we semantically enriched and extended CRM with concrete instances and more expressive structures, available only in OWL (like cardinality restrictions, inverse roles, existential and universal quantifications and so on). We then created a document named mondrian.owl that includes CRM concept and role instances which model facts from the life and work of the Dutch painter Piet Mondrian. In this document we also included axiom and fact declarations that OWL allows to be expressed, as well as new roles and concepts making use of this expressiveness.

The following code is a fragment from mondrian.owl stating that a “Painting_Event” is in fact a “Creation_Event” that “has_created” “Painting” objects only:

```
<owl:Class rdf:ID="Painting_Event">
<rdfs:subClassOf  rdf:resource="&crm;E65.Creation_Event"/>
  <rdfs:subClassOf>
    <owl:Restriction>
<owl:onProperty rdf:resource="&crm;P94F.has_created"/>
<owl:allValuesFrom rdf:resource="#Painting"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<Painting_Event rdf:ID= "Creation of Mondrian's composition">
<crm:P94F.has_created rdf:resource= "#Mondrian's
composition"/>
</Painting_Event>
```

The above fragment is graphically depicted in the left part of Figure 2. “Creation of Mondrian’s Composition” (i_1) is an explicitly stated “Painting_Event” that

"has_created" (***R***) "Mondrian's composition" (***i₂***). Now, asking the KDI to infer "what is a painting?" it infers that i_2 is indeed a painting (right part of Figure 2), correctly interpreting the value restriction on role R.

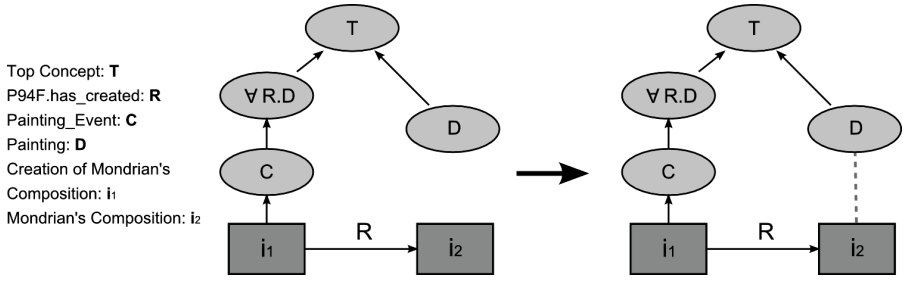


Fig. 2. Inference Example using Value Restriction

Let's now examine another example that involves the use of nominals. The following fragment from mondrian.owl states that a "Painting" is a "Visual_ Item" that its "Type" is "painting_composition".

```
<owl:Class rdf:ID="Painting">
<owl:subClassOf rdf:resource="&crm;E36.Visual_Item"/>
  <owl:equivalentClass>
    <owl:Restriction>
<owl:onProperty rdf:resource="&crm;P2F.has_type"/>
<owl:hasValue rdf:resource="#painting_composition"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
<crm:E55.Type rdf:ID="painting_composition"/>
<Painting rdf:ID= "Mondrian's composition"/>
```

The above fragment is graphically depicted in the left part of Figure 3. "Mondrian's Composition" (***i₁***) is explicitly declared as a "Painting" instance which in turn is defined as a hasValue restriction on "has_type" (***R***). "Painting_composition" (***i₂***) is declared as a "Type" object. While the fact that "Mondrian's Composition" "has_type" "Painting" is straightforward, the KDI is unable to infer so and returns null when asked "what is a the type of Mondrian's composition?"

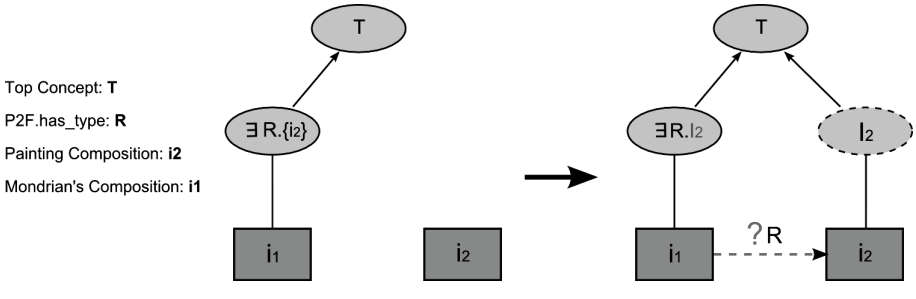


Fig. 3. Inference Example using Existential Quantification and Nominals

This example clearly demonstrates how difficult is for RACER as well as for every other current DL based system to reason about nominals. Given the $\{i_2\}$ nominal, RACER creates a new synonym concept I_2 and makes i_2 an instance of I_2. It then actually replaces the hasValue restriction with an existential quantifier on *concept* I_2 and thus is unable to infer that $R(i_1,i_2)$ really holds.

It is notable that proceeding with a DIG interface in this case would also not solve the problem, due to its known limitations. Clearly only a direct in-memory implementation, being interfaced by the appropriate OWL API, such as the ones provided by FaCT++ and Pellet would allow a successful answer to this kind of queries.

5 Conclusions

In this paper we have primarily argued about how a well-studied logical formalism, Description Logics, can be utilized in order to enable intelligent querying of Semantic Web documents. In order to achieve this, a key step was the review of available AI formalisms and system families that could be used to ground reasoning services upon. As the scene is currently set, DL-based systems appear to be the most promising choice to achieve streamlined inference results even in the short term. At the same time, DLs show adequate compatibility and corresponding systems tend to exploit the greatest part out of the Semantic Web ontological formalism expressiveness, as it is now standardized in OWL.

We believe that our hands-on experimentation with a number of state-of-the-art DL inference engines has produced at least two lessons learned: First, the need for instance-based reasoning, which we have shown to be of crucial importance for the Semantic Web environment [11], is now becoming *sine qua non* for the majority of the systems reviewed; second, we confirmed that there are still issues, even with the most advanced DL-systems, when trying to fully support OWL's decidable expressivity.

The potential as well as the limits of the DL-based approach are clearly demonstrated through our "wrapper prototype", the KDI: On the one hand, we have succeeded in demonstrating tangible and meaningful knowledge discovery results on Semantic Web documents, with a web-distributed architecture. On the other hand, we found that the KDI is greatly hampered by the difficulty of current DL inference engines to deal with nominals or, equivalently, by the absence of suitable communication standards and implementations. We trust though that at the near future most of the difficulties and incompatibilities identified throughout our work would be overridden by the evolution of systems and the refinement of the Ontology Web Language.

Acknowledgements

Dimitrios A. Koutsomitropoulos is partially supported by a grant from the "Alexander S. Onassis" Public Benefit Foundation.

References

1. Dickinson, I.: Implementation experience with the DIG 1.1 specification. Tech. Report HPL-2004-85, Hewlett Packard, Digital Media Sys. Labs, Bristol (2004)
2. Fikes, R., Jenkins, J., Gleb, F.: JTP: A System Architecture and Component Library for Hybrid Reasoning. In: Proc. of the 7th World Multiconference on Systemics, Cybernetics, and Informatics (2003)
3. Haarslev, V., Möller, R.: Racer: A Core Inference Engine for the Semantic Web. In: EON 2003. Proc. of the 2nd International Workshop on Evaluation of Ontology-based Tools, pp. 27–36 (2003)
4. Horridge, M., Bechhofer, S., Noppens, O.: Igniting the OWL 1.1 Touch Paper: The OWL API. In: OWLED 2007. Proc. of the OWL Experiences and Directions Workshop (2007)
5. Horridge, M., Tsarkov, D., Redmond, T.: Supporting Early Adoption of OWL 1.1 with Protégé-OWL and FaCT++. In: OWLED 2006. Proc. of the OWL Experiences and Directions Workshop (2006)
6. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 17–29. Springer, Heidelberg (2003)
7. Horrocks, I., Sattler, U.: A tableaux decision procedure for SHOIQ. In: IJCAI 2005. Proc. Of the 19th Int. Joint Conf. on Art. Intelligence, Morgan Kaufm., Seattle (2005)
8. Horrocks, I., Sattler, U.: Optimised reasoning for SHIQ. In: ECAI 2002. Proc. of the 15th Eur. Conf. on Artificial Intelligence, pp. 277–281 (2002)
9. Hsu, E., McGuinness, D.: Wine Agent: Semantic Web Testbed Application. In: Proc. Of Workshop on Description Logics (2003)
10. Kopena, J., Regli, W.C.: DAMLJessKB: A tool for reasoning with the Semantic Web. IEEE Intelligent Systems 18(3), 74–77 (2003)
11. Koutsomitropoulos, D.A., Fragakis, M.F., Papatheodorou, T.S.: Discovering Knowledge in Web Ontologies: A Methodology and Prototype Implementation. In: Proc. of SEMANTICS 2006, OCG, pp. 151–164 (2006)
12. Koutsomitropoulos, D.A., Papatheodorou, T.S.: Expressive Reasoning about Cultural Heritage Knowledge Using Web Ontologies. In: WEBIST 2007. Proc. of 3d Int.Conf. on Web Information Systems and Technologies, WIA track, pp. 276–281 (2007)
13. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics 5(2) (2007)
14. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, Springer, Heidelberg (2006)

Controlling Property Growth in Product Classification Schemes: A Data Management Approach

Joerg Leukel

University of Hohenheim, Information Systems II, Schwerzstr. 35 70599 Stuttgart, Germany
joerg.leukel@uni-hohenheim.de

Abstract. Product classification schemes aim at semantic interoperability in B2B e-commerce by providing consensual definitions of product categories and recommending properties for describing product instances. Considerable industry work has been carried out on enhancing the size and thus coverage of these schemes. Horizontal classification schemes, however, often consist of more than 10,000 classes, several thousand properties, and an even greater number of class-property relations. The problem is that maintaining these schemes becomes more and more demanding in particular due to the number of definitions and interrelations. This paper proposes measures for coping with the problem of extensive and steadily growing property libraries. We view these schemes from a data modeling perspective and relate the proposed measures to the underlying conceptual data model of product classification schemes. It can be shown that these measures greatly influence both standards makers and standards adopters.

Keywords: B2B, E-commerce, Product Data Management, Product Ontologies, Standardization.

1 Introduction

Executing business processes between independent organizations faces often heterogeneity concerning process models, data sources, software systems, and available meta data describing these components. Automating such processes increases the need for aligning heterogeneities and finding consensus about common concepts. Ontologies aim at fulfilling this role by establishing a shared and common understanding of a domain. In B2B e-commerce, most processes incorporate essential information about products (and services) being the subject of procurement and sales respectively. Therefore, the development of product ontologies can be regarded as an enabler of machine-readable, unambiguous representations of information about products [1] [2].

Many industry consortia have proposed such domain ontologies called standard product classification schemes (standard PCS). Applying these business vocabularies benefits searching for products in e-catalogs, comparing similar products, standardizing product descriptions, and facilitates spend analysis and product-sensitive workflows [3]. Prominent horizontal standards, such as eCl@ss, eOTD and UNSPSC consist of 20,000 up to 60,000 product classes, and represent a huge amount of knowledge about the categorization of products.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 363–374, 2008.

For standards makers, the broad coverage of horizontal standards leads to an enormous amount of properties as new subjects of standardization including proposal, negotiation, definition, and maintenance. Taking in mind the distributed nature of many standardization processes (e.g., work groups for each sub-domain or branch of industry), reducing or avoiding redundant properties becomes an important task. For instance, work groups should always check carefully the appropriateness of existing properties before proposing a new property for the vocabulary. This basic principle does not affect the underlying model, but the standardization process. It requires, however, that properties are reusable. Reusability of properties depends on their semantic precision, naming issues (i.e., synonyms, homonyms), and the conflict between wide or narrow definitions.

For standards adopters, properties must be seen from a different perspective. Classification based on a standard PCS requires (1) assigning each product to a class and (2) describing each product with class-specific properties. This initial effort is time-consuming and costing; it depends mainly on the number of products and the number of properties per class. In addition, it requires continuous efforts triggered by new PCS versions (new product classes, new properties, redefinition of classes and properties).

This paper proposes measures for coping with the problem of extensive and steadily growing property libraries. We view these schemes from a data modeling perspective and relate the proposed measures to the underlying conceptual data model of product classification schemes. Our contribution to research is that we (1) introduce the still overseen problem of property growth and (2) provide a comprehensive set of measures that address this problem.

The remainder of our paper is structured as follows. Section 2 discusses related work, and shows that extensive property libraries have rarely been the subject of research. In section 3, we describe the main problem based on empirical observations. In addition, we present the basic conceptual data model of property-centric classification schemes, which will be extended in the course of our work. In section 4, we define measures for coping with property libraries, and describe their impact on the problem. In section 5, we discuss the measures by summarizing their impacts and extending the basic model. Finally, we draw conclusions and point to future avenues of research.

2 Related Work

Related work to product classification schemes can be found in several fields such as e-catalogs, product data management, standardization, and ontology engineering. Next, we provide an overview of closely related work and outline their relevance to the problems caused by extensive property libraries.

Early work on basic concepts of PCS presents and evaluates standard PCS from a business perspective. For instance, [4] examined the role of standard PCS towards benefits of spend analysis; properties are not necessary for this business function.

Schulten et al. introduced product classification as a reference domain for ontology engineering and the Semantic Web, and called for concentrated efforts to “design a generic model” for automated mapping between two different PCS [5]. Concerning product properties, the proposal demonstrated the mapping problem between classes

only, but did not incorporate properties. Eventually, the research prototypes in [6] and [7] followed this class-centered path.

Ng et al. described challenges in integrating product schemes based on heterogeneous properties [8]. Property lists form schemas that can be integrated by applying techniques from database schema integration. Two interesting aspects discussed by Ng et al. are shallowness (flat structures, lists) and bushiness (clusters of a high number of related properties). This database approach is complemented and extended in [9], which analyzed property mappings and practical issues in more detail.

Leukel et al. proposed an XML-based exchange format for PCS [10]. Its contribution lies in identifying and defining data elements and relationships, both being derived from an empirical study of four standard PCS and three XML e-business standards. The modeling of properties is quite sophisticated and fulfills mainly requirements of PCS adopters. Two measures for supporting "management and maintenance" of property libraries – grouping and inheritance – are described briefly.

Recently, the importance of properties as a cornerstone of product classification has become more evident. For instance, Ondracek and Sander [11] argued on a "property based product classification" from that multiple different classification hierarchies for specific purposes can be built, though they are based on common, thus standardized properties. Leukel [2] emphasized the role of properties for providing additional semantics to class hierarchies; properties are needed to describe the scope of a class formally. Kim et al. developed a "semantic classification model" [12] based on properties in order to enable an in-depth understanding of product classification. All this work is in support of semantically rich PCS that incorporate well-defined properties. The problems caused by large property libraries are being overseen though.

A first indication of problems related to properties can be found in early work by Hepp [13]. The proposed quantitative measurements for PCS reveal some shortcomings in property lists and can help detect duplicate properties. In its conclusion, the paper argues on the need for further work on maintaining properties and organizing property libraries. A comprehensive quantitative analysis of classes and properties in PCS is subject of [14]. In addition, Ondracek and Sander drew attention to the problem of redundant properties in huge property libraries, and claimed that separating definition and application of properties would be "the only solution" [11].

3 Problem Definition: Property Growth

In recent years, considerable industry efforts have been undertaken to extend the semantic richness of product classes by adding class-specific property lists. A property list contains all properties that should be used to describe products belonging to the respective class. These lists greatly enhance the formal precision of standard PCS by replacing class labels with a structured, though human-language description of its meaning. From an ontology perspective, property lists can be regarded as a first step towards true product ontologies, since they provide standardized representations of product concepts, thus machine-readable semantics [14]. Defining property lists not only requires resources, but it also causes new problems for both standards makers and standards adopters. We refer to these problems as of *property growth*.

3.1 Observations

Property growth can be assessed by comparing multiple versions of the same standard PCS. Next, we present some data for eCl@ss, a horizontal PCS being developed by a consortium of mainly German companies since the late 1990s [15]. It has gained a significant relevance for e-procurement in many European countries. Table 1 shows basic figures for five versions of eCl@ss.

Table 1. Growth of Classes and Properties in eCl@ss

	V4.0	V4.1	V5.0	V5.1	V5.1.1
Publication Date	Aug 2001	Sep 2002	Dec 2003	Sep 2004	Sep 2005
4th Level Classes	10,190	12,565	20,379	21,100	22,203
Properties	2,303	5,504	3,667	5,525	6,941
Class-Property Relations	68,244	303,511	406,482	403,859	440,430

The number of classes, properties, and class-property relations has tremendously increased (118%, 201% and 545%) in the past four years. The increase of properties, however, is not constantly over time (significant decline in V5.0). We can assume that eCl@ss had implemented some actions to limit redundancy by reorganizing the property library, although, the sheer number of property remains high.

Looking closer at eCl@ss (table 2), we observe that property lists were added to more and more classes and most property lists contain at least 30 properties (V5.0: 83% of all property lists; V4.1: maximum of 294 properties!).

Table 2. Property Lists in eCl@ss

	V4.0	V4.1	V5.0	V5.1
Classes with Property Lists	1,107 (10.9%)	6,507 (51.8%)	7,913 (38.8%)	10,930 (51.8%)
Properties per Class: Minimum	6	6	3	1
Properties per Class: Maximum	89	294	266	156
Properties per Class: Mean	20.6	42.0	43.5	32.3
Properties per Class: Derivation	10.3	16.7	13.3	15.2

The growing number of property lists and properties per class causes significant classification costs, thus calls for property-centric classification strategies for PCS adopters (i.e., suppliers, buyers, marketplaces). Moreover, we have to consider that in many industries multiple standard PCS are available, especially due to competition between horizontal standards. Each standard PCS defines its own classes, properties, and property lists. Therefore, the problem of property growth is multiplied by the number of standard PCS.

3.2 Basic Model

Product classes are the core components of each PCS. A product class is a categorization, collection or type of similar products that share a set of characteristics (e.g., the

class ‘laptop’ describes portable computers). Product characteristics are expressed by properties (e.g., CPU type, display size, weight). Properties are not limited to a single class, but should be reusable. While some properties represent information that can be captured by standard data types (e.g., string, integer, float, Boolean), other properties limit the allowed values to a specific list of values (e.g., color ‘red’, ‘green’, ‘blue’ etc.). In conceptual modeling, the definition of enumerated domains can be expressed by a tertiary relation between product class, property and value as it is shown in figure 1. This data model introduces elementary attributes for each entity type. For instance, a property consists of its identification, name, textual definition, data type, and unit of measurement (UOM, e.g., meters, kilogram, and volt). Similar models are used in [9], [10], and [11].

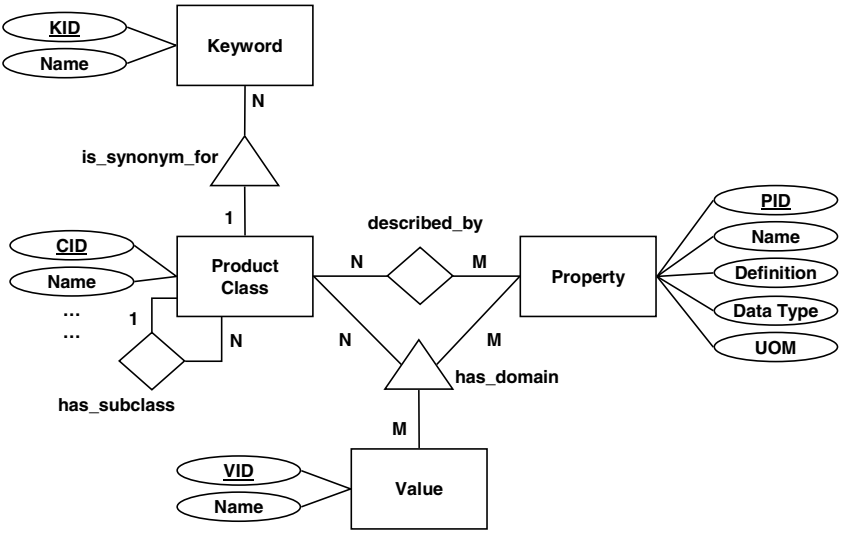


Fig. 1. Basic Conceptual Data Model

The major drawback of this model can be illustrated by a simple example: In industry segment A, which is represented by a number of classes, the property ‘length’ is measured in inches, while industry segment B – being represented by other product classes of the same sub-tree – measures the very same property in meters (e.g., hand tools vs. pipes for gas transportation). Consequently, two properties must be defined; their specification is nearly equal and differs only in UOM. This is especially for horizontal PCS, which cover a broad range of industry segments, a common problem. It can be solved by increasing the reusability of properties, though this may require modifications of the model.

4 Data Management Measures

In this section, we define a comprehensive set of measures that can be taken by standards makers. These measures address data management issues and relate to the underlying

conceptual data model. We describe the rationale and, if necessary, point out to extensions of the basic model.

For each measure we assess the potential on the given problem. There are 'two sides to every story' – standards must be developed and maintained, and standards should be applied; otherwise they would be no standards. The effect on standards makers concerns changes in initial efforts (i.e., setting up the PCS) and maintenance efforts (i.e., processing change requests, releasing new versions). Similarly, the effect on standards adopters refers to the initial classification process and subsequent re-classification processes. Another aspect of PCS adoption is the GUI presentation and actual usage in applications such as e-procurement, marketplaces, and product data management systems. This aspect will be considered, too.

4.1 Maximum Number of Properties

Rationale: The number of properties per class is limited to a fix number (e.g., 15); this limitation applies to all property lists, thus to the entire PCS. It prevents property growth locally, especially with regard to product segments in which product descriptions can be very detailed. This measure does not modify the basic model, but adds a constraint on the cardinality of the class-property relation.

Standards makers: The implementation of this measure requires making a single decision on the maximum number. Eventually, the size of the property library is limited as well (only by the number of classes and the reuse of existing properties). There is even a significant change in maintenance efforts, since adding a new property to a property list is not possible if the maximum number has already been reached.

Standards adopters: Both the classification and re-classification efforts are reduced and can be forecasted. The GUI representation is improved due to lower space requirements and may fit on a single screen in all cases (e.g., imagine the list of 294 properties compared to the reduced list of the 15 most important properties).

4.2 Optional Properties

Rationale: Properties are distinguished whether their usage is optional or mandatory. This distinction aims at reducing the number of essential properties that have to be used for product description, while it does not remove properties from property lists. The number of optional and mandatory properties depends on the product class; for instance, all properties may be optional, or mandatory (min/max-approach). This measure adds a new attribute 'mandatory' to the class-property relation.

Standards makers: For each property of each property list, the question of mandatory or optional has to be answered. This initial effort can be reduced by setting all properties to optional followed by searching for the most important, thus mandatory properties. Because of product innovation, optional properties may be shifted to mandatory and vice-versa (maintenance effort).

Standards adopters: Depending on the share of optional properties, the classification effort is reduced. This measure allows diverse classification strategies, i.e., support

only mandatory properties. Adopters decide on supporting optional properties, especially if these are required by their customers. Moreover, there are two consequences on GUI representation: (1) optional properties can be marked, thus separated from mandatory properties, and (2) parametric, property-based search for products has to be restricted to mandatory properties.

4.3 Naming Conventions

Rationale: The name of a property must adhere to specific naming conventions in order to prevent redundant properties (e.g., 'diameter, max.' vs. 'maximum diameter' vs. 'max. diam.'). This measure addresses the problem of finding the right property in the property library, in addition to keywords. Types of conventions: prefix vs. postfix qualifiers, singular vs. plural, use of abbreviations, separation of UOM from property name (e.g., 'diameter' instead of 'diameter in mm'). This measure does not modify the basic model.

Standards makers: Initially, naming conventions must be developed, and applied to all property names. Additional property names to the harmonized name can be stored in the keyword list. Applying these conventions can reveal redundant properties that should be removed from the property library.

Standards adopters: The classification process is not directly affected; searching for the right property is slightly improved. Since the naming can be used to build logical groups of properties, the GUI representation of large property lists is improved. For instance, postfixes to property names (e.g., 'length, max.', 'length, min.') express a specialization of closely related properties.

4.4 Property Groups

Rationale: Each property belongs to a predefined group (e.g., design, dimensions, shape, and business properties). This categorization eases the handling of huge property lists, since the flat list is transformed into a hierarchical structure. The basic model has to be modified: define a list of groups, and add a N:1 relation between property group to property.

Standards makers: Implementing this measure requires defining non-overlapping groups and assigning each property of the property library to one group. The maintenance effort is slightly effected (assign each new property to one group). The subdivision of the property library helps overlooking all properties, though it does not affect the total number of properties.

Standards adopters: Similarly to naming conventions, this measure does not influence the classification process. In the same way, it improves the GUI representation by explicitly defined groups of similar properties.

4.5 Views on Property Lists

Rationale: Instead of defining a single comprehensive property list, define overlapping views on property lists for each stage of the product lifecycle. The rationale is that the relevance of a property depends mainly on the product lifecycle and the

respective business function of product data. For instance, the requirements of spend analysis differ from those of engineering. In consequence, each view-specific property list can be reduced to purely relevant properties. Eventually, the class-property relation must be modified to reflect the view.

Standards makers: Views on property lists result in multiple, overlapping property lists for the same product class; hence the initial and maintenance efforts are considerably higher. While the number of properties in those lists is reduced because of including only view-relevant properties, the total number of properties in the property library remains unchanged.

Standards adopters: Due to strictly view-relevant properties, the efforts for classification can be reduced in those cases where not all product lifecycle phases are relevant. Often, standard PCS are only used for procurement or sales and not for intra-organizational purposes; therefore, this benefit is quite relevant. In addition, the GUI representation no longer contains non-relevant properties.

4.6 Property Templates

Rationale: Instead of defining properties completely, the property library contains templates only. These generic properties can be used for multiple specific purposes by concretizing the template. This concept is very similar to separating property definition and property application as described in [9]. For instance, the template includes name and definition, while the concretization adds data type and UOM. Regarding the basic model, the class-property relation is extended by further attributes that were formerly part of the property entity type.

Standards makers: The first step for implementing this measure is deciding which attributes still belong to the generic property and which attributes belong to the class-specific property, thus to the class-property relation. Eventually, a rather small number of generic properties needs to be defined from which more specific properties can be instantiated. Concerning maintenance, adding a new property can often be based on a similar, already existing property (i.e., concretizing the generic property instead of defining the new property completely).

Standards adopters: This measure does not influence the classification process nor does it improve the GUI representation. The reason is that property templates concern only the organization of the property library.

4.7 Property Inheritance

Rationale: So far, all measures were directed at properties only. Considering that properties are assigned to product classes forming a class hierarchy, property inheritance says that properties are inherited to all lower classes. Moreover, an inherited property can be modified (concretized) on lower levels. This measure does affect the basic model as follows: the class-property relation as well as the class-property-value relation is available for all classes, not only for leaves of the class tree.

Standards makers: Setting up a PCS based on property inheritance calls for thoroughly defined properties that can be assigned to nodes of the class tree; otherwise the

benefits of inheritance will not be realized. Moreover, the class hierarchy itself has to be suitable for assigning properties that are common for complete sub-trees. Maintaining such a PCS requires fewer efforts, since the property library contains lesser properties and sub-trees truly represent similar product classes characterized by a set of common properties.

Standards adopters: Similarly to property templates, this measure concerns the property library only.

5 Discussion

In this section, we discuss the proposed measures by summarizing the expected effects and modify the basic model.

5.1 Summarization of Effects

Next, we compile the previously assessed effects of each measure on the problem of property growth (table 3). For both standards makers and adopters, we state expected changes regarding initial and maintenance efforts ('-' for decrease; '+' for increase, 'o' for no change). Effects on the total number of properties and GUI representation are further criteria of our assessment ('reduced' and 'improved' respectively).

Comparing the effects on standards makers and adopters, we have to state that 6 out of 7 measures increase the initial effort for standards makers, while classification efforts are reduced or remain unchanged; a reduction of the number of properties can be expected for four measures, while the GUI representation is improved by five measures.

Table 3. Effects on Standards Makers and Standards Adopters

	Standards Makers			Standards Adopters		
Measure	**Initial Effort**	**Main-ten-ance**	**Number of Properties**	**Initial Effort**	**Main-ten-ance**	**GUI**
Maximum Number	-	-	reduced	-	-	improved
Optional Properties	+	o	o	-	o	improved
Naming Conventions	+	o	reduced	o	o	improved
Property Groups	+	o	o	o	o	improved
Views on Prop. Lists	+	+	o	-	-	improved
Property Templates	+	-	reduced	o	o	o
Property Inheritance	+	-	reduced	o	o	o

5.2 Modification of the Basic Model

Since the proposed measures concern the definition of single properties being elements of a property library, the basic model for PCS needs to be modified. Therefore, we collect the modifications described before and alter the basic model as shown in Figure 2. The modifications include (1) adding one new attribute ('Mandatory'), (2) adding two new entity types (property group, view), (3) redefining one relationship ('described_by'), and (4) moving attributes from the property entity type to the described_by relationship (here: data type, uom).

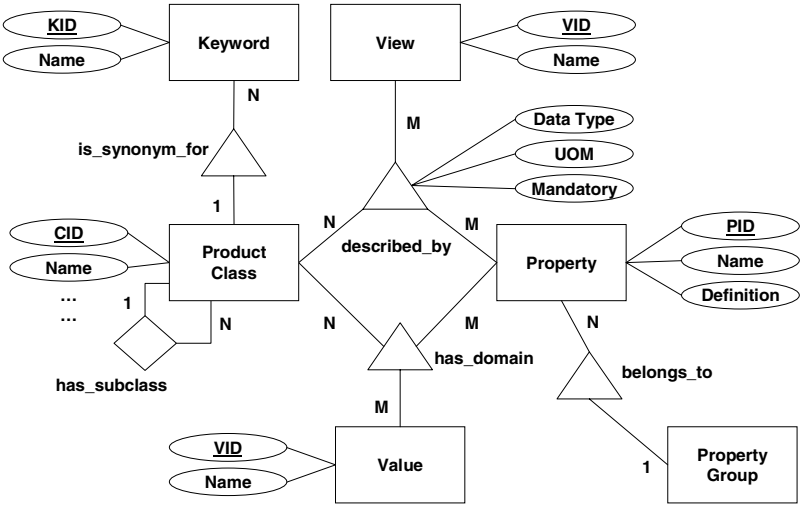


Fig. 2. Extended Conceptual Data Model

6 Conclusions

The main contribution of this paper lies in proposing new measures for coping with the problem of extensive and steadily growing property libraries. The assessment of their impact on standards makers and standards adopters revealed that the effects concern not only the total number of properties, but also initial efforts, maintenance efforts, and GUI representation issues. Therefore, decisions on implementing these measures should bear in mind all these criteria. The results of the assessment as well as the modified model may serve standards makers in their decision process on reorganizing property libraries.

The quantitative analysis (see section 3.1 and also [14]) of the property library in eCl@ss has drawn attention to the problem of defining, maintaining and actually using huge sets of product properties. While this quite elaborated standard PCS claims to be unique in its property-centered approach, the conceptual model of its property library is rather simple. We conclude that eCl@ss still focuses on semantic richness (i.e., extending the coverage of industry segments) rather than formal precision and efficient maintenance. This example, nevertheless, underlines the need for re-thinking the current organization of property libraries, since measures tackling the described problems are at hand, their impact can be predicted, and some of the proposed measures have already been tested in vertical standards.

Considering recent developments in finding consensus about the basic components, underlying conceptual models as well as maintenance policies of PCS, standards bodies and industry consortia have joined efforts in harmonizing their proprietary approaches in several initiatives and on different levels of obligation. For instance, the CEN project on product classification [3] states that a “good” PCS necessarily incorporates properties and property lists. Standard PCS that are purely based on classes

are expected to add properties in order to broaden the range of application and to provide extended semantics.

Concerning the current state of standard PCS, these transformation processes will be quite demanding. From this point of view, we plan future work on validating the measures by quantitative analysis of standard PCS, thus reengineering their property libraries based on automated conversion, and developing transformation strategies for standards makers. Another field of interest derives from the role and suitability of reference models for PCS. These models will become more important since many standard PCS aim at converting their proprietary data models to the ISO 13584 standard [16]. This standard requires some modifications to the property library, though it does not address the maintenance problem explicitly as described in this paper, since its main purpose is to achieve semantic interoperability between different property libraries. Therefore, we see the need for extending the scope of this reference model to content management issues that greatly determine the costs and efforts of defining and implementing respective standard PCS.

References

1. Fensel, D., et al.: Product Data Integration in B2B E-Commerce. IEEE Intelligent Systems 16, 54–59 (2001)
2. Leukel, J.: Standardization of Product Ontologies in B2B Relationships – On the Role of ISO 13584. In: 10th Americas Conference on Information Systems, AIS, pp. 4084–4091 (2004)
3. CEN/ISSS: CEN Workshop Agreement 15925:2005 – Description of References and Data Models for Classification. Brussels (2005)
4. Fairchild, A.M., de Vuyst, B.: Coding Standards Benefiting Product and Service Information in Ecommerce. In: 35th Annual Hawaii International Conference on System Sciences, p. 258b. IEEE Computer Society, Los Alamitos (2001)
5. Schulten, E., et al.: The E-Commerce Product Classification Challenge. IEEE Intelligent Systems 16, 86–89 (2001)
6. Corcho, O., Gómez-Pérez, A.: Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings. In: Workshop on E-Business and Intelligent Web at the 17th International Joint Conference on Artificial Intelligence (2001)
7. Beneventano, D., et al.: A web service based framework for the semantic mapping amongst product classification schemas. J. of Electronic Commerce Research 5, 114–127 (2004)
8. Ng, W., Yan, G., Lim, E.-P.: Heterogeneous product description in electronic commerce. ACM SIGeCom Exchanges 1, 7–13 (2000)
9. Bullig, A., Schnadhorst, T., Wilkes, W.: Mapping of product dictionaries and corresponding catalog data. In: 10th ISPE International Conference on Concurrent Engineering, pp. 225–234, Balkema Publishers (2003)
10. Leukel, J., Schmitz, V., Dorloff, F.-D.: A Modeling Approach for Product Classification Systems. In: 13th International Workshop on Database and Expert Systems Applications, pp. 868–874. IEEE Computer Society, Los Alamitos (2002)
11. Ondracek, N., Sander, S.: Concepts and benefits of the german ISO 13584-compliant online dictionary. In: 10th ISPE International Conference on Concurrent Engineering, pp. 255–262, Balkema (2003), www.DINsml.net

12. Kim, D., et al.: A semantic classification model for e-catalogs. In: 6th IEEE International Conference on E-Commerce Technology, pp. 85–92. IEEE Computer Society, Los Alamitos (2004)
13. Hepp, M.: Measuring the Quality of Descriptive Languages for Products and Services. In: Multikonferenz Wirtschaftsinformatik, pp. 157–168, Cuvillier, Goettingen (2004)
14. Hepp, M., Leukel, J., Schmitz, V.: A Quantitative Analysis of Product Categorization Standards: Content, Coverage, and Maintenance of eCl@ss, UNSPSC, eOTD, and the RosettaNet Technical Dictionary. Knowledge and Information Systems 13(1), 77–114 (2007)
15. eCl@ss e.V.: eCl@ss – Standardized Material and Service Classification, http://www.eclass-online.com
16. ISO: ISO 13584-1:2001 Industrial automation systems integration – Parts library – Part 1: Overview and fundamental principles. Geneva (2001)

Collaborative Visualization in Rich Media Environments

Bernd Eßmann[1,⋆], Frank Götz[2], and Thorsten Hampel[3]

[1] Heinz Nixdorf Institute, University of Paderborn
Fürstenallee 11, 33102 Paderborn, Germany
bernd.essmann@hni.upb.de
[2] University of Paderborn
Fürstenallee 11, 33102 Paderborn, Germany
frank.goetz@upb.de
[3] University of Vienna
Universitätsstrasse 10, A-1090 Wien, Austria
thorsten.hampel@univie.ac.at

Abstract. In this article we present our approach of combining open and sophisticated technologies in order to establish an integrated rich media environment for collaborative visualization processes. Aiming to support comprehensive visualization settings of spatially separated domain specialists, we deploy remote render farms for producing the visualization of complex datasets as video streams, separately for every collaboration partner. This makes our system capable also for low-end mobile devices, which only have to be able to render MEPG-4 compliant video streams. The cooperation support is provided by a full-featured CSCW system including a shared whiteboard based on the platform independent Eclipse framework. The visualization objects are embedded in the CSCW system's persistent object space and presented by the rich media view of the shared whiteboard. Starting with a basic scenario of collaborative visualization we will present the architecture of the combined visualization and CSCW systems and the design of the plug-in based shared whiteboard.

1 Introduction

Today's scientific communities are often distributed over several continents, which is emphasized by globalization. In cooperative research over long distances the coordination of research processes becomes a major challenge. This is especially true for research communities of domain specialists exploring complex datasets with the support of visualization techniques.

To identify the original incentive for the development of a cooperative visualization system we have to look at the past years. A few years ago the principal purposes of visualization systems were only the processing and the visual representation of complex datasets, which were generated from simulations or measurements. Analysis of the visualization and the underlying data happened at the same graphics workstation and at a distinguished spatial location.

⋆ Bernd Eßmann participated in the Heinz Nixdorf Institute's postgraduate program "Automatic Configuration in Open Systems" funded by the German Research Foundation (DFG).

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 375–387, 2008.

Nowadays, groups of domain specialists (often scientists) want to discuss and understand new geological phenomena cooperatively while being situated all over the globe. They want to access huge datasets (e.g. measurements of a geographical phenomenon) in real-time, independent from their actual whereabouts. A local computer has to process the dataset into a meaningful three-dimensional graphical representation. This allows domain specialists to get an overall understanding of the data. Additionally, for a cooperative exploration of the data, the domain specialists should be able to cooperatively navigate through the three-dimensional scene, annotate points of interest, or create snapshots of significant areas.

A common problem is that scientists have no access to graphic-workstations for generating and exploring the data locally. Furthermore, the necessary hardware is expensive and does not support the mobility of users. In contrast to these technical restraints, the trend of globally available network connectivity poses new potentials for solving this dilemma.

This paper presents our approach for real-time cooperation based on synchronous remote visualizations in a shared whiteboard application. Based on a scenario of a cooperation of spatially separated scientists (section 2), we present our conjunction of a *Computer Supported Cooperative Work (CSCW)* system and a remote visualization system by utilizing their open architectures. We introduce a shared whiteboard client based on the Eclipse framework embedding the visualization in an object-oriented manner (section 3). The paper closes with a presentation of related work and an outlook on future prospects.

2 Scenario of Use

Two spatial separated domain specialists (in our case geologists) try to understand the same climate phenomenon. Following the *old fashioned* way, the scientists would have to communicate sequentially, according to workflow specified earlier. This can be realized e.g. by email. In the beginning, a visualization expert processes the dataset into a meaningful representation. Then, the resulting representation will be delivered to the two geologists. Now, the representation is analyzed and discussed by the geologists. Additional enhancements will be made by a visualization expert. Again, the geologists will discuss the phenomena asynchronously via email. This procedure will be repeated until the geologists are satisfied with the results. Finally, the results have to be stored and distributed to other scientists. As a final step the scientists publish their results. They would have to manually create a web page including the pictures, annotations and any meta-data. Overall this suspended process is a very time consuming cooperation task.

Using our system the domain specialists and the visualization expert work in one shared workspace (see Fig. 1). The workspace is presented within a whiteboard, allowing graphical editing and annotating of the embedded objects. All objects are persistently stored on a CSCW server. Thus the representation in the shared whiteboard is persistent. The CSCW system is based on the metaphor of virtual knowledge spaces allowing the scientists to cooperate within a virtual room. A remote visualization system renders a three-dimensional representation of the dataset into an interactive object, created within the room and therefore shown in the shared whiteboard. This visualization

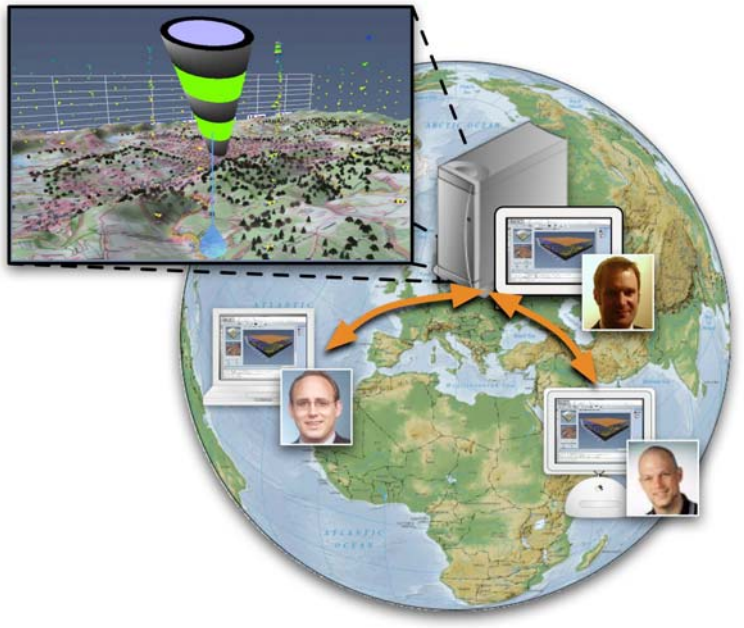


Fig. 1. Domain specialists distributed over the globe working on a shared visualization

object shows an interactive video stream delivered by a visualization cluster. The visualization object is visible as an interactive picture embedded in the shared workspace, which can be manipulated as any other object. Additionally, the visualization object may be attached with control panels for manipulating the visualization. These include a navigation pane, a moderation pane, and a preference pane.

It is possible to create snapshots from the actual scene and store them in the shared workspace. Measurements of single data entities within the dataset can be selected and stored as a cooperation object. These features help the domain specialists to exchange their ideas while working with the representation. The exchange itself can be realized by storing interesting visual bookmarks to the scene and annotating them.

An embedded chat facility derived from the CSCW system allows communication while cooperatively exploring the visualization. Additionally, to coordinate the exploration of the shared visualization, users may use the moderation function, which allows reserving timeslots for exclusive navigation in the scene. The results of the visualization process may be published on the fly in the form of a website at any time of the process. For this purpose the CSCW server generates a website containing the objects in the shared workspace, e.g. the annotated screenshots. On a reload, the website changes dynamically as a result of the changes in the whiteboard.

Because, the clients receive a video-stream instead of the raw data, the security of the dataset against theft and spying is enhanced. Furthermore, cooperation based on confidential data can take place without the need of sending raw data to the cooperation partner.

3 Architectural Design

In our approach the collaborative functionality (user management, access control, off-visualization annotation, and structuring) is provided by the CSCW system sTeam, while *openVisaar* provides the visualization (for examples of visualization techniques refer to Fig. 2). We now outline our concepts for the cooperation in so-called virtual knowledge spaces and present the symbiosis of both systems.

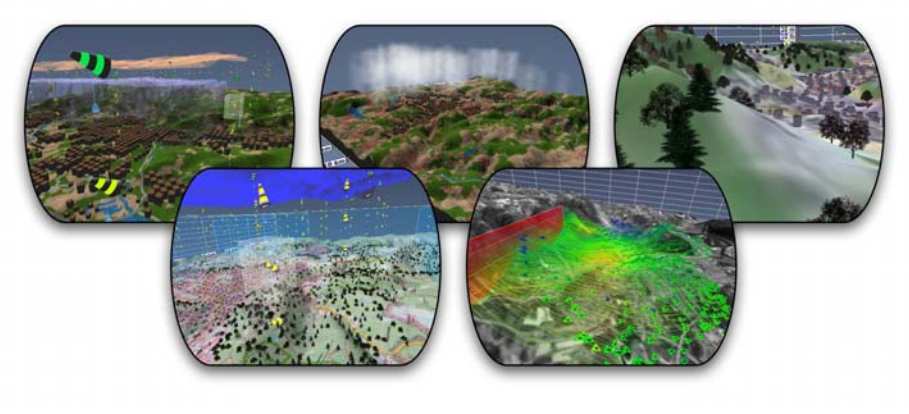


Fig. 2. Some visualization techniques provided by openVisaar

3.1 Embedding openVisaar in sTeam

Collaboration in Virtual Knowledge Spaces. *sTeam* is an open-source CSCW system developed at the University of Paderborn [1]. It provides a variety of flexible mechanisms to foster communication and cooperation in learning and work processes. sTeam's concept of *virtual knowledge spaces* combines synchronous and asynchronous forms of cooperation with hypermedia document management in flexible ways.

A sTeam server consists of a persistent object repository, which is stored in a relational database and a core that manages the access of the cooperation objects. In case of a change clients will be notified by an event system. This allows them to react to changes of any object in the cooperation system directly. Rights management for accessing contained objects is provided via *Access Control Lists (ACLs)*, which allow for flexible access right structures.

So far, sTeam provides no facilities for cooperative visualizations. Because of its extendibility and flexibility it was chosen as the CSCW basis for the approach presented in this paper.

Distributed Visualization with openVisaar. The novel visualization objects are embedded as a video stream within the sTeam whiteboard client. The remote rendering and video streaming is realized with the openVisaar system [2]. openVisaar is an *OpenSG* based visualization framework. Whereas, OpenSG is a portable scene graph system for creating real-time graphics programs using *OpenGL* [3]. openVisaar can multicast

in real-time rendered three-dimensional visualizations as *MPEG-4* video streams using the *Real Time Streaming Protocol (RTSP)* [4]. The whole visualization process is coordinated by the openVisaar server.

openVisaar is divided into a server and a client [5]. The server part consists of a cluster, composed of powerful computers equipped with up-to-date graphics accelerator boards, and appropriate main memory. Whereas, the client part is hosted on the users' devices. These can be standard PCs, laptops, or handhelds (see Fig. 3). The visualization of the data itself is rendered in the same way on all rendering nodes, but it is possible to choose between different views or synchronized views. The only requirements for the client computer are the ability to decode ISO-compliant MPEG-4 video streams in real-time and Java support.

An openVisaar server consists of different services that are executed on a cluster:

- *SceneServer (openVisaar Server):* Both, the SceneServer and SceneRenderer use OpenSG for their scene graph management and rendering. The OpenSG data structures are set up in a way that allows multiple independent threads to manipulate the scene graph independently without interfering with each other. This feature allows for synchronizing the manipulations of each user with the manipulations of other users. Finally, every user of the collaborative working community gets the same view on the current dataset.
- *SceneRenderer (Render Node):* Every SceneRenderer contains a replicated scene graph that will be regularly harmonized with the scene graph of the SceneServer. Every modification by other users is displayed immediately. One SceneRenderer exists for every remote client. The SceneRenderer generates an individual view of the shared visualization scene and sends a video stream using RTSP (Real Time Streaming Protocol) to the remote client.

Symbiosis of sTeam and openVisaar. In sTeam's virtual knowledge space the visualization objects are treated as any other cooperation object. With the appropriate view component the web interface as well as the synchronous whiteboard client are able to present the visualization object as a video stream. The controls for manipulating the visualization scene on the openVisaar server are provided as plug-ins additionally. Every running visualization video plug-in in combination with its controls is registered as one conventional client by the openVisaar server.

To coordinate both systems, the openVisaar server connects to the sTeam server as a common synchronous CSCW client using sTeam's proprietary COAL protocol. Only one connection is established for all visualization clients. This helps to save bandwidth and communication overhead. The openVisaar server organizes the distribution of the visualization representations by managing render nodes. These send the resulting video streams to the client objects embedded in the sTeam system. Fig. 4 shows an overview of the complete architecture of the combined systems.

3.2 Next Generation Whiteboard *medi@rena* with Visualization Capabilities

The *medi@rena* is a standalone application for accessing the cooperative groupware and learning platform sTeam[1]. As a native application the whiteboard allows forms

[1] In our former publications the *medi@rena* application was called *WhiteboardNG*.

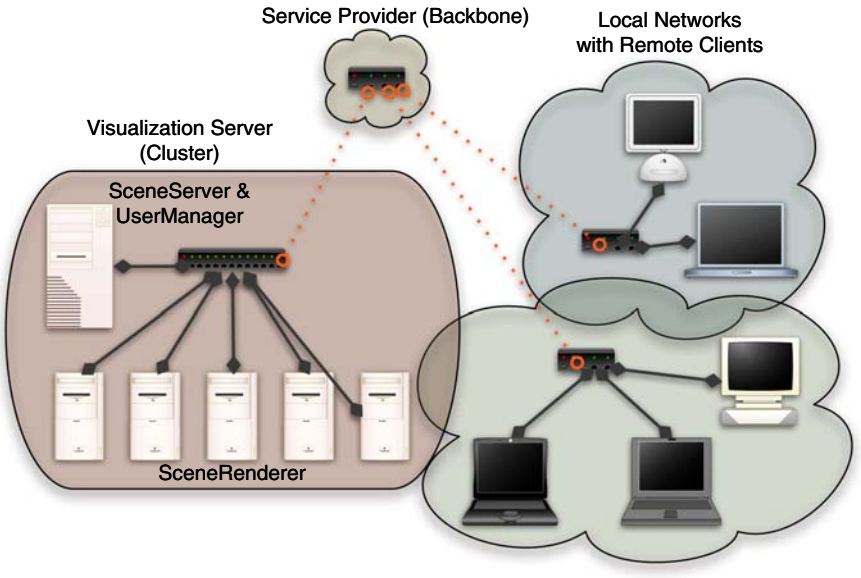


Fig. 3. The architecture of the *openVisaar* system

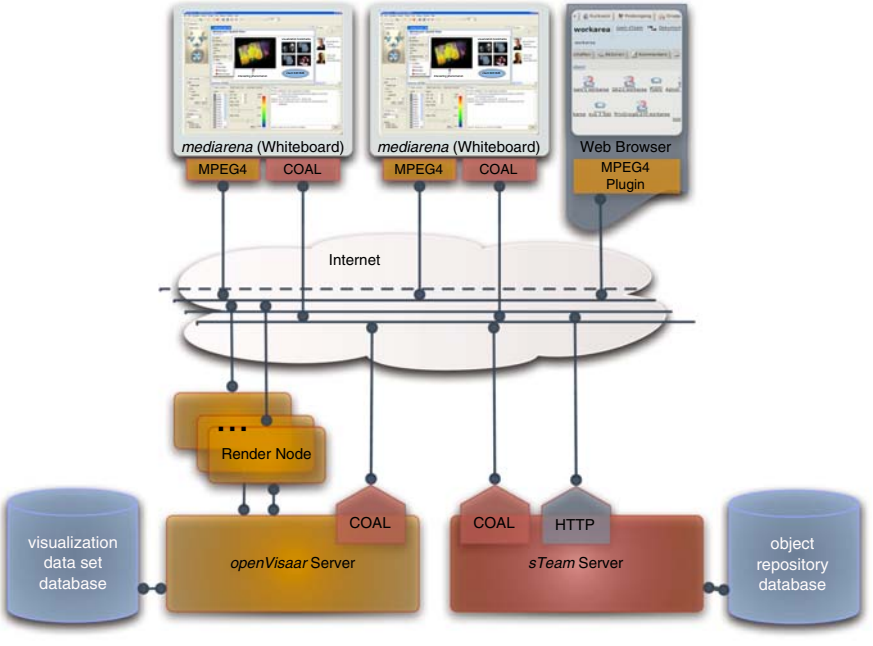


Fig. 4. Architecture of the integrative cooperative visualization system

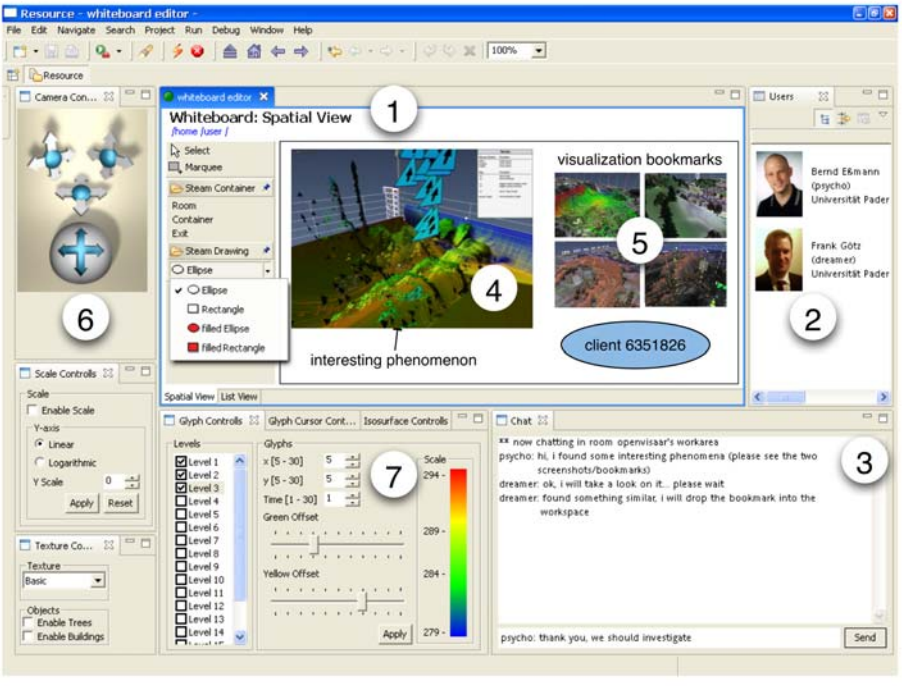


Fig. 5. The *medi@rena* with its standard components *whiteboard view* (1), *user view* (2), and *chat* (3), extended with the new visualization plugins (*visualization object* (4), *visualization bookmark objects* (5), *navigation control* (6) and *visualization preferences* (7))

of synchronous teaching [1] and offers improved interaction possibilities compared to pure HTML-based applications [6]. The main goal of the whiteboard is supplying an interactive and graphical view on the content of (areas in the) virtual knowledge spaces hosted on the sTeam server. Fig. 5 shows the medi@rena with its most important components. In the whiteboard area, objects like documents, collections, references, a trashcan etc. are displayed. Users can place and manipulate these objects individually within this area by using their mouse following the well-known workspace metaphor. Furthermore, objects can be annotated and spatially grouped by using graphical primitives such as rectangles, circles, arrows, lines etc.

New documents can be generated with the toolbar or they can be moved directly using the local file system (*drag & drop* in the workspace). In this way documents are persistently stored on the sTeam server.

Other parts of the User Interface (UI) are an *outline view* of available objects, a *miniature view* of the workspace, and a *user list* for mutual awareness of other users in the current area.

While a former implementation of the whiteboard application based on *SUN's Swing framework*, the novel medi@rena is based on *Eclipse* and uses parts of the Eclipse *Rich Client Platform (RCP)*. The sTeam clients functionality is realized by different plug-ins, which together form the entire application.

Additionally to the different components of the Eclipse platform the medi@rena uses functionality of the *Graphical Editor Framework (GEF)* and the *Eclipse Modeling Framework (EMF)* [7]. GEF provides methods and interfaces for developing *graphical editors* and EMF is a framework for modelling and managing complex data structures. Here, EMF is used for handling a proxy model of the sTeam server's data within the whiteboard. In addition to these two components, medi@rena consists of several plug-ins, which make sTeam's specific features available. Three specific components are of special importance in extending the medi@rena with new functionality:

- *Whiteboard.RCP:* This plug-in extends basis components of the Eclipse RCP, in order to build a medi@rena executable as an independent application. Therefore, in addition to the actual definition, product configuration, compilation of all plug-ins (used by the application), optical adjustments of icons, pictures and texts in information boxes are necessary. In the Eclipse jargon this process is called *Branding*. We use this component to generate a modified *visualization edition* of the whiteboard.
- *Whiteboard.Core:* This plug-in provides all non-visual kernel functions of the whiteboard. Among other things, the central classes *ModelManager* and *ConnectionManager* are part of this plug-in. For the integration of the visualization, specific core plug-ins provide access to both server systems.
- *Whiteboard.UI.Editor:* This plug-in implements (utilizing the GEF plug-in) the graphical editor with which a user can access and work interactively in a sTeam area. This plug-in has a special role within the whiteboard; it offers its own extension-point. With this extension-point third party developers have the possibility to implement own extensions for the editor of the whiteboard. These extensions are called edit-parts. An edit-part is the graphical representation of an object within a sTeam area.

  The visualization object is implemented as such an edit-part displaying the video stream provided by the openVisaar render nodes and providing a direct interaction with the visualization scene. Similar handlers exists for pictures, text documents, folders, lines, rectangles, arrows, etc.

The novel visualization component extends the existing whiteboard component of the medi@rena (see Fig. 6) with the functionality to display and control the visualization scene provided by the openVisaar server. At the same time, it enables the whiteboard to handle visualization objects like any other sTeam object in the knowledge space. This approach allows a seamless integration of cooperative visualization objects in the existing cooperation environment.

Platform Independent MPEG-4 Video Integration. For video streaming enabling visualization on the whiteboard, we chose the *MPEG-4 standard* [8] which is also used by the conventional openVisaar client. MPEG-4 (an extension of the MPEG-2 technology) is an ISO standard combining high-quality video streaming with minimal data rates. We use H.264 encoding, also known as MPEG-4 part 10, because it contains a number of new features that enable much more effective video compression and provides more flexibility for applications in a wide variety of network environments [9].

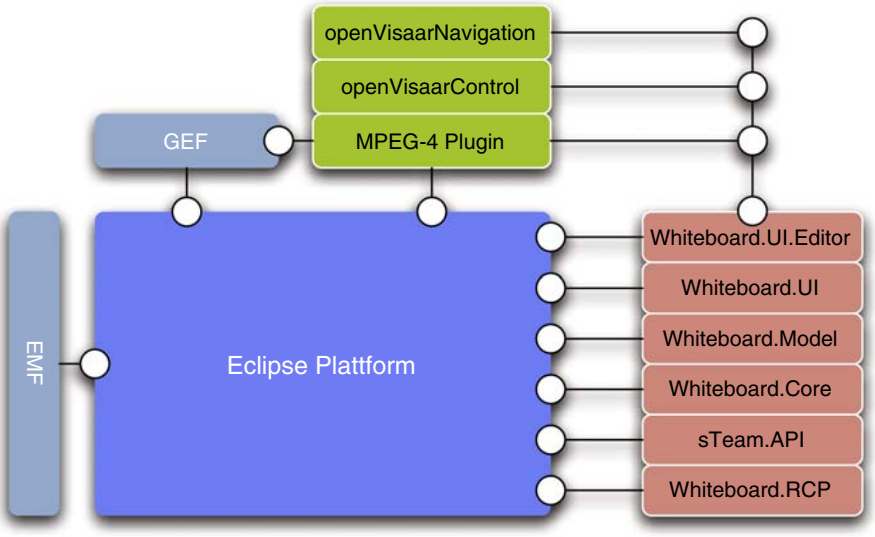


Fig. 6. The architecture of the Eclipse-based *medi@rena* extended with the novel visualization plug-in

As an open industry standard, anyone can create an MPEG-4 player or encoder. openVisaar uses the open source codec *XviD* for the encoding of video stream and the open source *MPEG4IP streaming server* called *MP4Live* [10] for streaming to the clients. On the client side, the video is seamlessly integrated in the sTeam whiteboard as a Java and GEF component avoiding media discontinuities in the cooperation process and allowing direct interaction with the visualization scene.

The standalone version of the openVisaar client uses *Apple Quick Time SDK* [11] for integrating video streaming into the client's user interface. Apple's Quick Time SDK provides good support for high quality MPEG-4 decoding with low CPU usage, but it is based on platform dependent runtime libraries. It was chosen, because *Sun's Java Media Framework* platform for independent video streaming lacks support for efficient streaming protocols like MPEG-4.

In our new approach the *IBM Toolkit for MPEG-4* [12] fills this gap. It is based on pure Java code and is therefore platform independent. Plus, it provides decoding facilities fast enough for our purposes. The drawback is the missing support of off-screen rendering, which is necessary to nest video streaming into GEF components. This leads to the problem that in addition to the video, rendered into GEF components, there is always a source window providing the decoded picture. While no other solution is available at the moment, this window is moved to the background or minimized after creation. Additionally, Quick Time stays integrated as an alternative rendering technique.

The Multimedia Content Description Interface. Our approach stores the metadata in two separate places: The *visualization storage* contains the data the visualization is based on (the scene graph managed by the openVisaar SceneServer); the *cooperation*

storage contains all data that is related to the collaboration process. The distinction is based on the premise that the first one stores object data, provided by measurement and monitoring; the second one stores annotations in the form of analysis, additional an related data, partly subjective and cross media, which can be attributed to specific subjects and persons.

The cooperation storage must therefor be capable to keep the annotation data that is created during the collaborative process. On a sTeam based system, this data will be stored as a sTeam object. To increase interchangeability with other systems and to provide an easy export of sequences as video in conjunction with all relevant annotation data we chose *MPEG-7* [13] for storing the annotations. The *Multimedia Content Description Interface MPEG-7* is an XML based standard for describing multimedia content developed by the *Moving Pictures Experts Group (MPEG)*. It consists of a set of description schemes and descriptors to define metadata elements and their relationships. Because MPEG-7 descriptions do not depend on the way the described content is encoded or stored, it provides a universal annotation approach for all types of media to facilitate searching, indexing, filtering, and accessing. For our purpose, MPEG-7 features will be used for storing the metadata generated by annotations of snapshots or sequences that where previously created from the visualization scene.

Adaptations to the openVisaar Client. The openVisaar client was originally implemented as a standalone client, using Sun's Swing components to provide the user interface. First, the user interface had to be adapted to the *Eclipse Standard Widget Toolkit (SWT)*. Here, the separation of the user interface code and the functional code in the openVisaar client application proved useful. A redesign of some parts containing the code for managing connections to the openVisaar server and handling remote events was necessary, too. These parts supported one-to-one connections only, as the client operated in a Single Document Interface (SDI) like manner. In conjunction with sTeam, one workspace may contain multiple openVisaar objects. Therefore multiple connections based on the same environment must be supported. Finally, we used the extension-point mechanism of the Whiteboard.UI.Editor to extend the *medi@rena* with a new edit-part for displaying the visualization. A new edit-part handles the graphical representation of our openVisaar based visualization object within a sTeam area.

4 Related Work

Existing annotation systems for *video annotation* provide capabilities only for persistent media and within environments where annotations are shared asynchronously. These tools concentrate on functionalities and features aiding the semantic labeling of video databases to organize authoring processes and assist in searches or analyses of specific snapshots.

The *E-Chalk Tool* [14], allows users to perform collaborative work on a virtual blackboard. In a typical setting, a lecture or meeting room is equipped with a projection system, a touch sensitive whiteboard or a digitizer tablet, and a computer with an internet connection. The E-Chalk server transmits all written data to the virtual blackboard. Audio comments and video of the lecture room are sent via internet connection to remote

participants. Only a Java compatible browser is needed to receive the audio, video, and the board image previously recorded. However, E-Chalk does not allow clients manipulating or presenting own content during a session. It also does not support even basic collaborative functions as found in environments like Microsoft's NetMeeting, where users can communicate by chat or voice, while working on the same media content in real time.

Like E-Chalk, *IMC's Lecturnity tool* [15] offers the possibility of presenting multimedia content for meetings, e-learning, e-manuals, and software training. Therefore it supports synchronous recording of audio and video in combination with screen-grabbing and annotation features. Based on PowerPoint presentations or running applications as a showcase, all mouse movements and clicks are recorded. Once the recording is finished, the author can edit and optimize the content. After a presentation is arranged in structured learning modules, it can be published on a CD-ROM or in the world wide web. The architecture of Lecturnity focuses on the fast creation and easy publishing of presentations, but like the E-Chalk tool lacks any functionality for collaborative work.

Despite their missing features for cooperation purposes, the presented annotating systems allow recording of freehand drawings and replaying them. The way they link annotations to video material is always based on time stamps and overlay positions in the video stream, because the source for the video material is no longer editable once recorded. This is different in remote visualization systems, where the source data is computed on the visualization servers on the fly. This method allows linking annotations directly to primitives of the scene that is rendered into the video stream. Finally, we want to look at some cooperative visualization systems.

OpenGL Vizserver from *SGI* [16] is a commercial client-server system. It is designed to deliver visualization and collaboration functionality to any client, whether on a desktop workstation or a wireless handheld computer. OpenGL Vizserver allows users to remotely view and interact with large datasets from any other system at any location within an organization in a cooperative manner.

The *CoVis (learning through collaborative visualization)* project was finished 1998 at the *Northwestern University*. A principal purpose of the project is the use of new technologies for the extension and improvement of the learning process. The visualization tools can be started from the network, but deliver no direct mechanisms for collaborative work. Regarding the paradigms of virtual knowledge spaces it lacks flexible and expandable structures for the cooperation process. Since CoVis consists of a collection of individual tools, media discontinuities evolve by the missing integration of individual applications [17].

Habanero, which was developed at the *Software Development Division at the National Center for Supercomputing Applications (NCSA)*, offers a Java-based framework (-architecture) for the production of cooperative environments. Habanero is session and tool oriented and offers tools like a whiteboard, telnet, or an audio chat. Because of a missing concept for integrating the tools into a common cooperation environment, data exchanges between them are not possible out of the box. This is a major disadvantage for flexible cooperation settings [18].

Generally speaking, the presented cooperative visualization systems deliver interesting solutions for some aspects of the collaborative visualization process. In our opinion,

however with regards to a flexible concept of cooperation support, virtual knowledge spaces are the most promising concept for the desired cooperative visualization environment.

5 Outlook

By combining the visualization system openVisaar with the CSCW system sTeam, new ways of analyzing and discussing complex visualizations in teams evolve. While openVisaar provides sophisticated visualization techniques presenting all relevant data to the participants adequately, sTeam serves as a platform supporting the overall collaboration process. One key element of sTeam is the shared whiteboard, allowing synchronous cooperative work in a graphical manner. Different media types are represented by graphical objects generated, grouped, or generally structured during the collaborative process.

The seamless integration of openVisaar into sTeam enables users to collaboratively browse, discuss, annotate, and publish results, even when working at different locations over long distances. Contrary to other media types (e.g. text documents, pictures, or pre-rendered videos), openVisaar generates volatile content by delivering real-time generated and streamed ISO-compliant MPEG-4 video. Although, sTeam supports tools for collaborative work on the above-mentioned immutable media types. These are inapplicable in several cases: when the lifetime of media is limited to the actual session, where its creation and representation is based on specific parameters, and where knowledge is gained from the media by comparison of content changes, rather than concrete states.

The communication between the openVisaar server and sTeam is based on a simple, but proprietary protocol. Future switching to a protocol like *SOAP* might open the involved systems even more to other systems.

Concluding, the solution presented in this paper surely outlines the benefits and synergy effects gained by combining open systems in order to create novel and highly integrated forms of collaboration in rich media environments.

References

1. Hampel, T., Keil-Slawik, R.: sTeam: Structuring Information in a Team - Distributed Knowledge Management in Cooperative Learning Environments. ACM Journal of Educational Resources in Computing 1 (2), 1–27 (2002)
2. Goetz, F., Domik, G.: Remote and Collaborative Visualization with openVisaar. In: Proceedings of the 3rd IASTED International Conference on Visualization, Imaging, and Image Processing, pp. 902–907 (2003)
3. Reiners, D., Voss, G., Behr, J.: OpenSG – Basic Concepts. In: Proceedings of the 1st OpenSG Symposium (2002)
4. Goetz, F., Domik, G.: A Framework for Video-based and Hardware-Accelerated Remote 3D-Visualization. In: Proceedings of the Eurographics 2003 Short Presentations, pp. 263–269 (2003)
5. Goetz, F., Domik, G.: openVisaar - Enriching cooperative visualization by combining open technologies. In: Brueckner, S.A., Di Marzo Serugendo, G., Hales, D., Zambonelli, F. (eds.) ESOA 2005. LNCS (LNAI), vol. 3910, pp. 38–46. Springer, Heidelberg (2006)

6. Hampel, T., Eßmann, B.: Self-administered cooperative knowledge areas - evaluation of the www interface in terms of software ergonomics. In: Proceedings of the HCI International 2003, pp. 729–733 (2003)
7. Moore, W., Dean, D., Gerber, A., Wagenknecht, G., Vanderheyden, P.: Eclipse Development using the Graphical Editing Framework and the Eclipse Modeling Framework. IBM Press (2004)
8. Koenen, R.: MPEG-4 Overview. ISO/IEC JTC1/SC29/WG11 N4668 (2002)
9. De Neve, W., Lambert, P., Lerouge, S., Van de Walle, S., Van de Walle, R.: Assessment of the Compression Efficiency of the MPEG-4 AVC Specification. In: Proceedings of SPIE/Electronic Imaging 2004, vol. 5308, pp. 1082–1093 (2004)
10. Mackie, D.: Streaming Video and MPEG4IP. In: Presentation of MPEG4IP at the Silicon Valley Linux User's Group, Cisco Technology Center, Silicon Valley, USA (2002)
11. Apple Computer, Inc.: Quick Time for Java (2005), http://developer.apple.com/
12. IBM alphaworks, Gr.: IBM Toolkit for MPEG-4 (2005), http://www.alphaworks.ibm.com/
13. Martinez, J.M.: MPEG-7 Overview. ISO/IEC JTC1/SC29/WG11 N4980 (2002)
14. Friedland, G., Knipping, L., Schulte, J., Tapia, E.: E-Chalk: A Lecture Recording System using the Chalkboard Metaphor. Journal of Interactive Technology and Smart Education 1 (1), 9–20 (2004)
15. Mueller, R., Ottmann, T.: The Authoring on the Fly system for automated recording and replay of (tele)presentations. Multimedia Systems 3 (8), 158–176 (2000)
16. Silicon Graphics, Inc.: OpenGL Vizserver 3.1: Application-Transparent Remote Interactive Visualization and Collaboration. Technical White Paper (2005), http://www.sgi.com
17. Ramamurthy, R.K.B., Wilhelmson, R., Pea, R.D., Gomez, L.M., Edelson, D.C.: CoVis: A national science education collaboratory. In: Proceedings of the American Meteorological Society 4th Conference on Education joint with the 11th Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, pp. 15–20 (1995)
18. Chabert, A., Grossman, E., Jackson, K., Pietrovicz, S.: NCSA Habanero - Synchronous collaborative framework and environment. White Paper, Software Development Division at the National Center for Supercomputing Applications (1998)

A Formal Semantics and a Client Synthesis for a BPEL Service

Serge Haddad[1], Patrice Moreaux[2], and Sylvain Rampacek[3]

[1] LAMSADE, Université Paris Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
haddad@lamsade.dauphine.fr
[2] LISTIC-Polytech'Savoie, Université de Savoie
BP 80439, 74944 Annecy le Vieux Cedex, France
patrice.moreaux@univ-savoie.fr
[3] LE2I, Université de Bourgogne
BP 47870, 21078 Dijon Cedex, France
sylvain.rampacek@u-bourgogne.fr

Abstract. A complex Web service described with languages like BPEL4WS, consists of an executable process and its observable behaviour (called an abstract process) based on the messages exchanged with the client. The abstract process behaviour is non deterministic due to the internal choices during the service execution. Furthermore the specification often includes timing constraints which must be taken into account by the client. Thus given a service specification, we identify the synthesis of a client as a key issue for the development of Web services. To this end, we propose an approach based on (dense) timed automata to first describe the observable service behaviour and then to build correct interacting clients when possible. The present work extends a previous discrete time approach and overcomes its limitations.

Keywords: Web Services, BPEL4WS, Algebra of timed processes, Timed Automata.

1 Introduction

From Elementary Web Services to Complex Ones. Web services are "self contained, self-describing modular applications that can be published, located, and invoked across the Web" [1]. They are based on a set of independent open platform standards to reach a high level of acceptance. Web services framework is divided into three areas: communication protocol, service discovery and service description. The "Web Services Description Language" (WSDL) [2] provides a formal, computer-readable description of Web services. Such a description specifies the software component interfaces listing the collection of operations that are network accessible through standard XML messaging. It includes all information that an application needs to invoke such as the message structure, the response structure and some binding information like the transport protocol, the port address, etc. However simple operation invocation is not sufficient for some kind of composite services. They require in addition a long-running interaction

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 388–401, 2008.

derived by an explicit process model. This kind of services may often be encountered in two cases. First when a Web service is developed, it is composed by a set of accessible operations and a process model which schedules the invocation to a correct use of the service. Secondly, facing to the capability limits of Web services, composite services may be obtained by aggregating existing Web services in order to create more sophisticated services (and this in a recursive way). In order to deal with the behavioural aspects of complex services, some industrial and academic specifications languages have been introduced. Among them, "Business Process Execution Language for Web Services" (BPEL4WS or more succinctly BPEL) has been proposed by leading actors of industry (BEA, IBM, and Microsoft) and has quickly become a standard [3].

The Two Facets of Complex Web Services. BPEL supports two different types of business processes - see for instance [4], [5]:

- Executable processes specify the exact details of business processes. They can be executed by an orchestration engine.
- Abstract business protocols specify the public message exchange between the client and the service. They do not include the internal details of process flows but are required in order for the client to correctly interact with the service.

Given the description of an executable process, its associated interaction protocol is obtained by an abstraction mechanism (which masks all the internal operations of the service). However the issues raised by these two types of processes are very different. A specification of an executable process is close to the definition of a program whereas the specification of interaction protocol mainly raises an difficult problem: how to synthetize a client which will correctly handle the interaction with the service.

The Synthesis Problem. Indeed by construction, the external behaviour of a service is non deterministic due to its internal choices. It is then *a priori* unclear whether a client, i.e. a deterministic program, can be designed to interact with it. Furthermore the specification often includes timing constraints (e.g. implicit detection of the withdrawal of an interaction by the client) implying that these timing constraints must also be taken into account by the client. However since no semantics of the interaction process is given for BPEL (not to be confused with the semantics of the service execution), this problem could not be formally stated.

A Discrete Time Solution [6,7]. In this work, the authors have specified what is an external behaviour, i.e. they have given an operational semantics to an abstract BPEL specification in terms of a discrete time transition system. The semantics is obtained by a set of rules in a modular way. Given a constructor of the language and the behaviour of some components, a rule specifies a possible transition of a service built via this constructor applied on these components. As previously discussed, the transition system is generally non deterministic.

Then they have defined a relation between two communicating systems which formalizes the concept of a correct interaction. There are standard relations between dynamic systems like the language equivalence and the bisimulation equivalence but none of them matches their needs. Thus they have introduced the interaction relation which

can be viewed as a bisimulation relation modified in order to capture the nature of the events (i.e. the sending of a message is an action whereas the reception is a reaction).

Afterwards they have focused on the synthesis of a client which is in an interaction relation with the transition system corresponding to the system. The client they look for must be implementable, in other words it should be a deterministic automaton. It has appeared that some BPEL specifications do not admit such a client i.e. they are inherently ambiguous. Thus the algorithm they have developed either detects the ambiguity of the Web service or generates a deterministic automaton satisfying the interaction relation. The core of this algorithm is a kind of determinisation of the transition system of the service.

Our Present Contribution. In the previous solution, the discrete time semantics is preferred for simplicity reasons. However such a solution has the following drawbacks. First, the passing of a unit of time is modelled by an explicit transition in the transition system which means that the compact representation of timing constraints by values is now hidden in the model by their combination with logical transitions. In other words, whereas handling correctly the interaction with the service, the client automaton is hardly understandable by a user. Moreover if two timing constraints are not of the same order, the time unit must be chosen w.r.t. the shorter one leading to a combinatory explosion of the automaton due to the "translation" of the longer one.

Here we develop a dense time semantics for a BPEL specification as a timed automaton [8]. A preliminary version of this semantics for XLANG [9] was given in [10]. Switching from XLANG to BPEL is relatively easy thanks to the way the operational semantics is defined. The construction of this automaton is based on modular rules which works similarly to the discrete case. However they are more intricate since, on the one hand, the values of the timing constraints are handled symbolically with the help of clocks and, on the other hand, given some expression we must determine which clocks are active and how they govern the guards of the transitions. Next, we revisit the interaction relation. Contrary to the previous relation, we manage explicitly the time since now an execution is a discrete event sequence where the events are stamped by the time of their occurrence. The last part of the work is the most difficult one. Let us recall that the core of the former synthesis algorithm for the client automaton is similar to an automaton determinisation. It is well-known that the non deterministic timed automata are strictly more expressive than the deterministic ones. Thus we have designed an original procedure which decides whether a specification is (potentially) ambiguous and in the negative case produces such a deterministic automaton. Similarly to the approaches which determinise subclasses of timed automata [11], we require that the deterministic automaton has the same clocks as those of the original one.

Related Work. The platform WSAT [12,13] enables designers of a Web service composition to model check properties expressed by LTL formulas with SPIN tool. The formal semantics is obtained by gluing patterns for each BPEL construction. One pattern is connected from its final state to the initial state of next pattern according to the BPEL description with local transitions. This work does not cover the time features and it focuses only on message exchanges: the conversation is obtained by a *virtual watcher* that is supposed to record all messages sequences sent by each peer enrolled in the composition.

Another research of Web services formal semantics is based on a BPEL to Finite State Processes (FSP) translation [14]. This work lies on message sequence charts and the core of the verification mechanism consists to check trace equivalence. Again, the time features of the specification are not taken into account.

[15] uses the notation CRESS (Chisel Representation Employing Systematic Specification) to formalise Web services. This model presents two main advantages: automatic translation into formal languages for analysis as well as into implementation languages for deployment. Then the CRESS specification is translated into LOTOS and analysed with tools like TOPO, LOLA and CADP. Again, the temporal aspects are not present.

These different contributions share with our approach the design of a formal semantics for Web services. However they study the BPEL execution process and not the interaction protocol, they do not include the time features of BPEL and they perform component verification whereas we perform component synthesis.

Organization of the Paper. In section 2, we describe how to associate a formal semantics with a business process leading to the building of a timed automaton. Section 3 presents the client-service interaction relation. The synthesis of a client automaton is described in section 4. We summarize our results in section 5 and we present some directions for future works.

2 A Formal Semantics for BPEL Abstract Processes

BPEL provides a set of operators describing in a modular way the observable behaviour of an abstract process. As shown in [16], this kind of process description is close to the process algebra paradigm illustrated for instance by CCS [17], CSP [18] and ACP [19]. However, time is explicitly present in some of the BPEL constructors and thus the standard process algebra semantics are inappropriate for the semantics of such a process.

Thus our semantics associates a timed automaton (TA) with an abstract process. Let us briefly describe what is a TA. A TA is a (non deterministic) finite automaton enhanced with a finite set of clocks (C). An execution of a TA consists of (possibly null) time steps interleaved with discrete transitions. A TA configuration is composed of a state and a value per clock (called a clock valuation). The discrete transitions correspond to the edges between the states. The clocks control the behaviour of the TA as follows. An elementary constraint about a clock x, is $x < v$, $x \leq v$,$x > v$, $x \geq v$ where v is an integer constant. A constraint is a conjunction of elementary constraints. The set of constraints is denoted Γ. A constraint called an invariant is attached to each state. Time can elapse in a state if the invariant associated with the state is satisfied during all the successive configurations. In addition to its label, the specification of an edge also includes a constraint, here called a guard, and a subset of clocks to be reset. In order to follow this edge from a configuration, the corresponding guard must be true and the clock valuation after the reset operation must satisfy the invariant of the new state.

Definition 1 (Timed automaton [8]). *A Timed Automaton (TA) is a tuple* $T = (L, C, A, Inv, E, l_0)$ *where*

- L *is the set of* locations *or control states,*
- C *is the set of* clocks*,*

- A *is the set of* actions,
- Inv *a mapping from* L *to* Γ *defines the invariants,*
- $E \subseteq L \times \Gamma \times A \times 2^C \times L$ *is the set of* edges. *An edge* e *is* (s, g, a, r, d) *with* s *the source location,* g *the guard of* e, a *the action,* r *the subset of clocks reset by* e *and* d *the destination location.*
- l_0 *is the initial location.*

2.1 The Alphabet of the TA

The first step for the defining a semantics consists in specifying the action alphabet for a BPEL process. We have four kinds of actions:

- Silent actions, denoted by τ cannot be observed by the client. They correspond to decisions taken by the server (evaluation of a condition for switch, while, etc.). The exception events set of BPEL4WS is denoted by E_x.
- Timeout expirations which are denoted by .
- Sending and receiving messages: the set of types of messages will be denoted by M. The emission is denoted by $!m$ and the reception is denoted by $?m$. We also set $!M = \{!m \,|\, m \in M\}$ and $?M = \{?m \,|\, m \in M\}$ and the wildcard $*$ may be substituted for ! or ?.
- In order to control that the client correctly detects the end of the service, we introduce $\surd$, the termination event. This action will also simplify the definition of the operational semantics.

2.2 The States and the Edges of the TA

Each state will be associated with a BPEL process obtained by successive transformations from the initial process. Two states have different associated processes. At the beginning of the construction, there is a single state (the initial one) corresponding to this process. Each time an edge is defined, a new process is computed and if this process does not label an existing state then such a state is created. Due to the semantic rules given in the next subsection, it can be proved that the number of derived processes is finite (and thus the number of states is also finite).

The edges starting from a state are obtained by a top down analysis of the process expression labelling this state. This analysis is usually defined with the help of operational semantic rules. The definition of a semantic rule $[op_x]$ for a generic process $P = op_x(P_1, P_2, \ldots)$ is given as for Algebra of Timed Processes [20]. Components of a rule are:

- A boolean expression over some potential transitions of selected components of P: , $(\{P_{o(i)} \xrightarrow{\alpha_i} P'_{o(i)}\})$;
- This condition is enforced by a second condition on the occurring labels denoted by , $(\{\alpha_i\})$.
- If the two conditions are fulfilled then a state transition for P is possible where the label , $(\{\alpha_i\})$ is an expression depending on the labels of subprocesses transition and

– The new process is an expression $\cdot, (P, \{P'_{o(i)}\})$ depending on the original process and the new subprocesses.

So, a generic rule, presented with the usual style has the following structure:

$$[op_x] : \frac{Bexp(\{P_{o(i)} \xrightarrow{\alpha_i} P'_{o(i)}\})}{P \xrightarrow{Lexp(\{\alpha_i\})} Nexp(P, \{P'_{o(i)}\})} \text{ where } guard(\{\alpha_i\})$$

The guard and reset associated with an edge will be defined in the next subsection. For sake of readability, we do not follow the (verbose) XML syntax of a BPEL process. Instead we have chosen a simplified syntax close to the one used for process algebra whose meaning should be immediate for who knows BPEL. As usual, we begin the definition of rules by giving the ones corresponding to the basic processes of BPEL. These basic processes are `empty`, $?o[m]$, $!o[m]$ and `throw`$[e]$.

The `empty` **process** `empty` can only terminate (the notation 0 is the null process).

$$\texttt{empty} \xrightarrow{\surd} 0$$

The $?o[m]$ **and** $!o[m]$ **processes** The process $?o[m]$ (which corresponds to the input operation of WSDL) consists in receiving a message of type m. The process $!o[m]$ (which corresponds to the notification operation of WSDL) consists in sending a message of type m. We consider only these two types of WSDL operations. The two other types can be built with the sequence constructor (see below).

$$*o[m] \xrightarrow{*m} \texttt{empty} \quad \text{with} * \in \{?, !\}$$

The `throw` *Process.* The process `throw`$[e]$ raises an exception e which must be catched in some `scope` process.

$$\forall\, e \in E_x,\ \texttt{throw}[e] \xrightarrow{e} 0$$

The `sequence` *Process (* ; *).* The process $P\ ;\ Q$ executes the process P then the process Q. Since the operator ";" is associative, we safely restrict the number of operands to two processes. The `sequence` process acts as its first subprocess while this process does not indicate its termination. In the latter case, the `sequence` process acts as the second process can do.

$$\forall\, a \neq \surd,\ \frac{P \xrightarrow{a} P'}{P\ ;\ Q \xrightarrow{a} P'\ ;\ Q}$$

$$\forall\, a,\ \frac{P \xrightarrow{\surd} \text{ and } Q \xrightarrow{a} Q'}{P\ ;\ Q \xrightarrow{a} Q'} \text{ where } a \in \{!m, ?m, \surd, \tau\}$$

Note that if there is an action $a \neq \surd$ such that $P \xrightarrow{a} P'$, then $P \xrightarrow{\surd}$ cannot occur.

The `switch` *Process.* The process `switch`$[\{P_i\}_{i\in I}]$ chooses to behave as one process among the set $\{P_i\}$. Each branch of its execution is guarded by an *internal* condition.

Conditions are evaluated w.r.t. the order of their appearance in the description. However since the client has no way to predict the choice of the service, this order is irrelevant. The main consequence is that from the point of view of the client, *this choice is non deterministic*. The `switch` process becomes one of its subprocesses in a silent way. Let us note that we have implicitly supposed that at least one condition is fulfilled. In the other case, it is enough to add the process `empty` as one of the subprocesses.

$$\forall\, i \in I,\ \texttt{switch}[\{P_i \mid i \in I\}] \xrightarrow{\tau} P_i$$

The `while` *Process.* The process $\texttt{while}[P]$ iterates an inner process as long as an *internal* condition is satisfied. Like `switch`, `while` evaluates in a silent way its condition. Thus we have two rules depending on this internal evaluation.

$$\texttt{while}[P] \xrightarrow{\tau} P\ ;\ \texttt{while}[P]$$

$$\texttt{while}[P] \xrightarrow{\tau} \texttt{empty}$$

The `flow` *Process.* The process $\texttt{flow}[\{P_i\}_{i\in I}\}]$ simultaneously activates a set of processes $\{P_i\}$. For the moment considering that the synchronization primitives of BPEL are internal ones we have not yet implemented this synchronization. Thus this parallel execution is similar to a "fork-join" in the sense that the combined process ends its interaction when all subprocesses have completed their execution. Subprocesses of a `flow` process act independently except for one action: they simultaneously indicate their termination. In the latter case, the `flow` process becomes the null process. Furthermore internal actions are considered as immediate and consequently the occurrence of such an action in a subprocess prevents the occurrence of a delayed action (sending or reception of a message) in another subprocess.

• Individual actions:

1.
$$\frac{\begin{array}{c}\forall\, a \in E_x \bigcup \{\tau\},\\ \exists\, j \in I,\ P_j \xrightarrow{a} P'\end{array}}{\texttt{flow}[\{P_i \mid i \in I\}] \xrightarrow{a} \texttt{flow}[\{P_i \mid i \in I\backslash\{j\}\} \bigcup \{P'\}]}$$

2.
$$\frac{\begin{array}{c}\forall\, m \in M,\\ \exists\, j\in I,\ P_j \xrightarrow{*m} P' \text{ and}\\ \forall\, i \neq j,\ \forall\, a \in E_x \bigcup \{\tau\},\ \text{not } \exists k \in I,\ (P_i \xrightarrow{a})\end{array}}{\texttt{flow}[\{P_i \mid i \in I\}] \xrightarrow{*m} \texttt{flow}[\{P_i \mid i \in I\backslash\{j\}\} \bigcup \{P'\}]}$$

• Termination:

$$\frac{\forall\, i \in I,\ P_i \xrightarrow{\surd} P_i'}{\texttt{flow}[\{P_i \mid i \in I\}] \xrightarrow{\surd} 0}$$

The `scope` *Process.* $\texttt{scope}(P, E)$ with

$$E \stackrel{def}{=} [\{(m_i, P_i) \mid i \in I\}, (d, Q), \{(e_j, R_j) \mid j \in J\}]$$

may evolve due to P evolution, reception of a message m_i, expiration of the timeout with duration d or occurrence of an exception e_j. We note $M_I = \{m_i \,|\, i \in I\}$ and $E_J = \{e_j \,|\, j \in J\}$.

• P actions: The termination exits the scope whereas another action does not.

$$\frac{P \xrightarrow{\surd}}{\texttt{scope}(P,E) \xrightarrow{\surd} 0} \qquad \frac{P \xrightarrow{a} P'}{\texttt{scope}(P,E) \xrightarrow{a} \texttt{scope}(P',E)}$$

• Receiving a message m_i:

$$\forall\, i \in I,\ \frac{\forall\, a \in E_x \bigcup \{\tau, \surd\},\ \neg(P \xrightarrow{a})}{\texttt{scope}(P,E) \xrightarrow{?m_i} P_i}$$

• Exception handling: which depends whether the raised exception is catched in this scope.

$$\forall\, j \in J,\ \frac{P \xrightarrow{e_j}}{\texttt{scope}(P,E) \xrightarrow{\tau} R_j}$$

$$\forall\, e \notin E_J,\ \frac{P \xrightarrow{e}}{\texttt{scope}(P,E) \xrightarrow{e} 0}$$

If an exception e is never catched at any level then the process is an erroneous one which can straightforwardly checked by examining whether an exception labels an edge of the TA.

The `pick` *Process.* can be viewed as a particular case of the `scope` process.

2.3 The Clocks of the TA

We associate a clock with each `scope` process and a special clock (x_{im}) for handling the immediate actions. Given a process, we determine by a top down analysis which clocks are active, i.e. which `scope` subprocesses are activated. The invariant associated with a state depends whether an immediate action is possible. If it is the case, the invariant is $x_{im} = 0$ else the invariant is a conjunction over the active clocks of elementary conditions $x \leq d$ where d is the value defined in the scope corresponding to x.

The clocks to be reset when following an edge are simply the clocks which were inactive in the source process and become active in the target process. x_{im} is always reset.

There is no guard for the transitions defined by the operational rules. However, we add to each state which owns active clocks, a set of edges labelled by to (one per subset of active clocks which can simultaneously reach their bound). For each such edge the guard specifies that these active clocks have reached their bound while the other ones have not.

2.4 The Overall TA Construction

The computation of the timed automaton of the service can now be summarized as follows.

- It manages a set of processes to be examined and a current version of the TA. It starts with the initial process and a automaton reduced to a single state.
- When examining a process, it first builds the edges corresponding to the operational rules and for each target process not already present in the TA, it adds it to the set of processes to be examined.
- Then it determines the set of active clocks of the current process. Based on this information and the previous edges, it determines the invariant of the state. Afterwards, it generates the "time-out" edges.
- The information updating about the clock reset may take place at two different moments. If an edge points to an already encountered state then (based on the active clocks of the two states) this information is immediately updated. Otherwise, it will be updated when the target state will be examined.

3 Interaction Relation

It should be clear that the TA is a compact description of the observable behaviour of the BPEL process. However, as briefly sketched above, TA have themselves a formal semantics defined in terms of a Timed Transition System (TTS). A TTS over the set of actions A is a tuple $(S, s_0, A, \rightarrow)$ where S is a set of states, $s_0 \in S$ is the initial state, A is a finite set of actions disjoint from time passing, and $\rightarrow \subseteq S \times (A \cup \mathbb{R}_{\geq 0}) \times S$ is a set of edges. If $(q, e, q') \in \rightarrow$, we also write $q \xrightarrow{e} q'$. $q \xrightarrow{d} q'$ with $d \in \mathbb{R}_{\geq 0}$ corresponds to d units of time passing. The states of the TTS associated with a TA are simply the reachable configurations and its edges are either its discrete transitions or time passing in a location.

We first informally what should be a correct interaction between two TTS. As for the bisimulation relation, we require a relation between pairs of states of the two systems. Obviously the pair consisting of the initial states should belong to this relation.

Furthermore, the states of a pair should have a coherent view of the next interaction steps to occur. At first, this implies that the relation must take into account the mutually observable steps. Thus we introduce the observable transition relation of a TTS by $s \stackrel{a}{\Rightarrow} s'$ iff $s \stackrel{\tau^* a \tau^*}{\rightarrow} s'$, $s \stackrel{\epsilon}{\Rightarrow} s'$ iff $s \stackrel{\tau^*}{\rightarrow} s'$ and $s \stackrel{d}{\Rightarrow} s'$ iff $s \stackrel{d_1 \tau \ldots \tau d_n}{\rightarrow} s'$ with $\sum d_i = d$.

Once it is done, we could require (like for bisimulation) that if a state s of the pair (s, s') may evolve by an observable transition of its TTS to some new state s_1, s' should have a similar observable transition leading to a state s'_1 which would compose with s_1, a new pair of consistent views.

However we need to be careful. First, if a TTS sends a message the other one must be able to receive the message. So it is necessary to introduce the notion of complementary actions $\overline{?m} = !m$, $\overline{!m} = ?m$ and $\forall a \notin \{!m\}_{m \in M} \cup \{?m\}_{m \in M}\ \overline{a} = a$ and to require that the synchronized evolution is obtained via complementary actions.

But this requirement is too strong as it does not capture the different nature of the sending and reception of a message. A sending is an action whereas a reception is a

reaction and will not spontaneously occur. Therefore a more appropriate relation will first require that if, in s belonging to the pair (s, s'), a TTS may receive a message m, then there is a third state s'' of the other TTS indistinguishable from s' w.r.t. the observable transitions which can send m and second that in s' the other TTS can send a message (not necessarily m). The first condition expresses that the former TTS is not over specified and the second one that it will not wait indefinitely for a message.

These considerations yield the following formal definition.

Definition 2 (Interaction relation). *Let* $A_1 = (S, s_{01}, A, \rightarrow_1)$ *and* $A_2 = (S, s_{02}, A, \rightarrow_2)$ *be two TTS. Then* A_1 *and* A_2 correctly interact *iff* $\exists \sim \subseteq S_1 \times S_2$ *such that:*

- $s_{01} \sim s_{02}$
- $\forall s_1, s_2$ *such that* $s_1 \sim s_2$
  - *Let* $a \notin \{?m \mid m \in M\}$*; if* $\exists\, s_1 \stackrel{a}{\Longrightarrow}_1 s_1'$*, then* $\exists\, s_2 \stackrel{\overline{a}}{\Longrightarrow}_2 s_2'$ *with* $s_1' \sim s_2'$ *and if* $\exists\, s_2 \stackrel{a}{\Longrightarrow}_2 s_2'$ *then* $\exists\, s_1 \stackrel{\overline{a}}{\Longrightarrow}_1 s_1'$ *with* $s_1' \sim s_2'$
  - *Let* $m \in M$*; if* $s_1 \stackrel{?m}{\Longrightarrow}_1 s_1'$ *then*
    - $\exists\, s_2^- \stackrel{w}{\Longrightarrow}_2 s_2$, $\exists\, s_2^- \stackrel{w}{\Longrightarrow}_2 s_2^+$, $\exists\, s_2^+ \stackrel{!m}{\Longrightarrow}_2 s_2'$ *with* $s_1 \sim s_2^+$ *and* $s_1' \sim s_2'$ *where* w *is a word*
    - $\exists\, s_2 \stackrel{!m'}{\Longrightarrow}_2 s_2'$
  - *Let* $m \in M$*; if* $s_2 \stackrel{?m}{\Longrightarrow}_2 s_2'$ *then*
    - $\exists\, s_1^- \stackrel{w}{\Longrightarrow}_1 s_1$, $\exists\, s_1^- \stackrel{w}{\Longrightarrow}_1 s_1^+$, $\exists\, s_1^+ \stackrel{!m}{\Longrightarrow}_1 s_1'$ *with* $s_1^+ \sim s_2$ *and* $s_1' \sim s_2'$ *where* w *is a word*
    - $\exists\, s_1 \stackrel{!m'}{\Longrightarrow}_1 s_1'$

4 Client Timed Automaton Synthesis

We are now in position to present the client synthesis algorithm. First as the client must be implementable, we require that it is *deterministic*. Second as it must handle clocks to manage on its side the timeout of the service, we need to express its behaviour by a TTS. These consideration lead to choose as model for our client a deterministic timed automaton. Last, we must produce a TA which is in interaction relation with the TA of the BPEL process.

Before developing we emphasize that there exist BPEL process which do not admit clients. For instance, process $\texttt{switch}[?o[m], ?o[m']]$ internally chooses to receive either a message m or m' and thus no deterministic TA can correctly interact with it since it would imply that, in its initial state, the client should send either m or m' while the server would wait the other message. Note also the difference with process $\texttt{switch}[!o[m], !o[m']]$ where a client can be easily designed: just wait for either m or m'. We say that a process is *ambiguous* if it does not admit a deterministic TA which is in interaction relation with it.

4.1 The Synthesis Algorithm

Our algorithm does not look for any deterministic TA but restricts its search for a TA which has the same clocks as the TA of the BPEL process. Thus when the algorithm

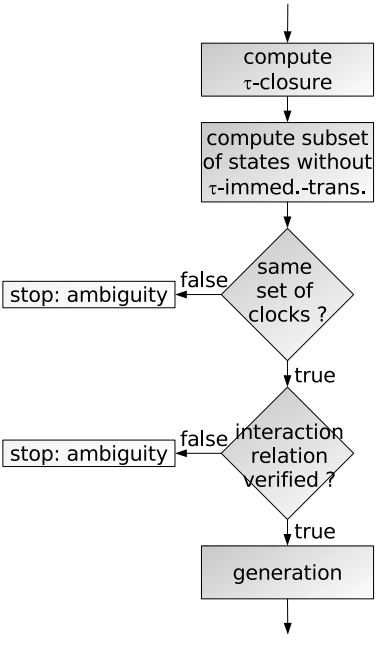


Fig. 1. A step of the client synthesis algorithm

outputs "ambiguity", it just means that no TA with this constraint exists. In other words, our procedure is not complete. However this restriction seems to be reasonable (see the next subsection about incompleteness of our algorithm).

The general principle of our algorithm (see figure 1) is similar to a determinisation procedure: a state of the TA client will correspond to a subset of states of the TA of the service (see fig. 2).

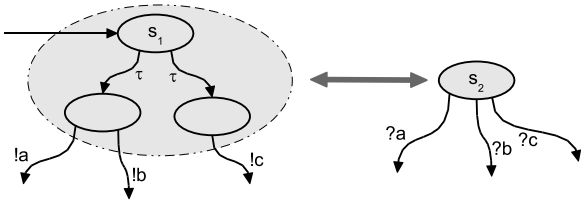


Fig. 2. Service subset of states (left) - client state (right)

More precisely, each potential state s of the TA client is associated with a subset of states $S_2(s)$ of the TA service which are related to s via the interaction relation. During the construction, there is a stack of client states to be processed. At the beginning of the algorithm, the stack contains an initial client state s_{01} such that $S(s_{01}) = \{s_{02}\}$,

s_{02} being the initial state of the service. It stops either when the stack is empty (i.e. the client has been built) or when it has detected the ambiguity of the service.

First, we compute the ϵ-closure by τ-transitions. If this subset (call it S') of service states is already associated with a state s of the client, then the edge of the client TA which has generated the subset is redirected to s. Otherwise, one creates a new client state (say $snew$).

We compute a subset of S' (say S'') that contains only states who has not output τ-immediate-transition.

Next, we compute the set of clocks for each state of S''. If this clock set is not unique, then there is an *ambiguity* (temporal ambiguity case) and we stop the construction. Afterwards, we check the interaction relation for discrete transitions. If it is not fulfilled then we also stop the construction.

The construction of the clocks guard has two steps. The first one consists to copy clock guards of the edges and clock invariants of the vertice of the TA server and, next, the complementary transitions are constructed.

4.2 Incompleteness of the Algorithm

As already discussed, our algorithm is incomplete. We give now an example of a false detection (see figure 3 corresponding to process `switch` $(!o[c], \texttt{scope}\ (!o[a], [\{(b, \texttt{empty})\}, (4, \texttt{empty}), \{\}])$): a process starting by a `switch` such that one branch of this `switch` starts with a `scope` process and another branch does not activate a timing constraint is detected as an ambiguous service. Indeed in one branch there is an active clock whereas in the other there is none.

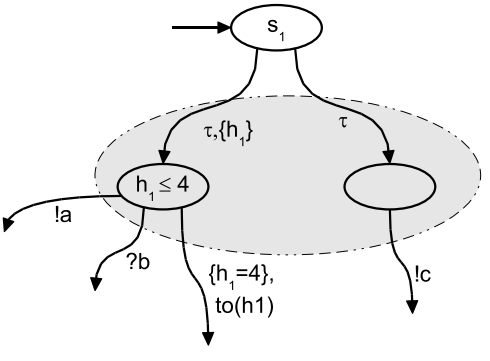


Fig. 3. A false detection of ambiguity

In a discrete time framework, the previous (complete) method [7] produces a client. Indeed the time elapsing is symbolized by an action (χ) and we implicitly work at a (discrete) TTS level. Here we work at a higher level (the TA one). Thus incompleteness is the price to pay in order to obtain a more compact representation of the client.

5 Conclusion

We have shown that the interaction with a Web service requires a theoretical development relative to its semantics. Extending a previous work in discrete time, we have proposed in this paper a dense time semantics for BPEL: from the definition of a service - an abstract process - we build a timed automaton corresponding to its formal semantics. Then we have defined an interaction relation between client and service considering them as timed transition systems. We have also designed an algorithm synthetising a deterministic timed automaton (the client) when this is possible and detecting ambiguous service otherwise.

This approach is implemented in our framework ICIS and will be soon reachable from the net. For sake of simplicity, we have considered a perfect communication channel (no loss and no delay). We are currently working on the generalization of our approach by including the specification of the channel characteristics.

References

1. Tidwell, D.: Web services - the web's next revolution. In: IBM developerWorks (2000)
2. WSDL: Web services description language (wsdl) 1.1. Technical report, World Wide Web Consortium (2001), http://www.w3.org/TR/wsdl
3. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business process execution language for web services (2003)
4. Juric, M.: BPEL and Java. On line journal theserverside.com (2005) http://www.theserverside.com/articles/article.tss?l=BPELJava
5. Juric, M., Sarang, P., Mathew, B.: Business Process Execution Language for Web Services. Packt Publishing (2005)
6. Melliti, T., Haddad, S.: Synthesis of agents for web services interaction. In: Workshop Semantic Web Services for Enterprise Application Integration and E-Commerce of the Fifth International Conference on Electronic Commerce, Pittsburgh, USA (2003)
7. Haddad, S., Melliti, T., Moreaux, P., Rampacek, S.: Modelling web services interoperability. In: ICEIS 2004. Proc. of the 6th Int. Conf. on Enterprise Information Systems, Porto, Portugal (2004)
8. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)
9. Thatte, S.: Xlang: Web services for business process design. World Wide Web page (2001), http://www.gotdotnet.com/team/xml/wsspecs/xlang-c/default.htm
10. Haddad, S., Melliti, T., Moreaux, P., Rampacek, S.: A dense time semantics for Web services specifications languages. In: ICTTA 2004. Proc. of the 1st Int. Conf. on Information & Communication Technologies: from Theory to Applications, Damascus, Syria, pp. 647–648. IEEE Computer Society Press, France (2004)
11. Alur, R., Fix, L., Henzinger, T.A.: Event-clock automata: a determinizable class of timed automata. Theoretical Computer Science 211(1–2), 253–273 (1999)
12. Fu, X., Bultan, T., Su, J.: Analysis of interacting bpel web services. In: WWW 2004. Proc. of the 13th International World Wide Web Conference, ACM Press, New York (2004)
13. Fu, X., Bultan, T., Su, J.: Wsat: A tool for formal analysis of web services. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, Springer, Heidelberg (2004)

14. Foster, H., Uchitel, S., Magee, J., Kramer, J.: Model-based verification of web service compositions. In: Proc. of the 18th Int. Conf. on Automated Software Eng. (2003)
15. Turner, K.J.: Formalising web services. In: Wang, F. (ed.) FORTE 2005. LNCS, vol. 3731, pp. 473–488. Springer, Heidelberg (2005)
16. Staab, S., van der Aalst, W., Benjamins, V., Sheth, A., Miller, J., Bussler, C., Maedche, A., Fensel, D., Gannon, D.: Web services: Been there, done that? IEEE Intelligent Systems 18, 72–85 (2003)
17. Milner, R.: Communication and Concurrency. Prentice-Hall, Englewood Cliffs (1989)
18. Hoare, C.: Communicating Sequential Processes. Prentice Hall, Englewood Cliffs (1985)
19. Bergstra, J., Klop, J.: Process algebra for synchronous communication. Information and Control 60(1-3), 109–137 (1984)
20. Nicollin, X., Sifakis, J.: The algebra of timed processes, atp: theory and application. Inf. Comput. 114(1), 131–178 (1994)

Part V

Human-Computer Interaction

Multi-modal Contact-Less Human Computer Interaction

Frangiskos Frangeskides[1] and Andreas Lanitis[2]

[1] Department of Computer Science, Cyprus College
PO Box 22006, Nicosia 1516, Cyprus
ffranges@cycollege.ac.cy
[2] Department of Multimedia and Graphic Arts, Cyprus University of Technology
P.O Box 50329, Lemesos 3066, Cyprus
andreas.lanitis@cut.ac.cy

Abstract. We describe a contact-less Human Computer Interaction (HCI) system that aims to provide paraplegics the opportunity to use computers without the need for additional invasive hardware. The proposed system is a multi-modal system combining both visual and speech input. Visual input is provided through a standard web camera used for capturing face images showing the user of the computer. Image processing techniques are used for tracking head movements, making it possible to use head motion in order to interact with a computer. Speech input is used for activating commonly used tasks that are normally activated using the mouse or the keyboard. The performance of the proposed system was evaluated using a number of specially designed test applications. According to the quantitative results, it is possible to perform most HCI tasks with the same ease and accuracy as in the case that a touch pad of a portable computer is used.

1 Introduction

Conventional Human Computer Interaction (HCI) relies on the use of hands for controlling the mouse and keyboard thus effective HCI is difficult (and in some cases impossible) for paraplegics. With our work we aim to design a system that will enable paraplegics to use a computer system. The proposed system is a multi-modal system that combines both visual input and speech input in order to allow the user to achieve full control of a computer system, in a hands-free fashion. Visual input is provided through a standard web camera attached on the monitor of the computer. Images showing the user of the system are analyzed in order to track his/her head movements. The face tracker activates cursor movements consistent with the detected head motion allowing the user to control cursor movements using head motion. Visual input can also be used for activating mouse clicks and entering text using a virtual keyboard. Figure 1 shows users using a computer system based on the system developed in this project.

The proposed system incorporates the capability of performing certain HCI tasks based on speech input. Speech signals used by the system, are captured using standard microphones attached to the computer or using headsets with microphones. Speech input can be utilized in two different modes of operation: The Sound Click Mode and the Voice Command Mode. When using the Sound Click Mode, speech input is used

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 405–419, 2008.

Fig. 1. Hands-Free HCI based on the proposed system

only as a means for activating a mouse click. In this case the user only needs to generate any sound in order to activate a click. In the Voice Command Mode we use speech recognition techniques so that the user can verbally request the execution of predefined tasks. Verbal commands handled by the system have been carefully selected in order to minimize the possibility of speech recognition errors and at the same time allow the user to carry out usual HCI tasks efficiently. The algorithms developed as part of the project, formed the basis for developing a prototype hands-free HCI software package. The package contains a program that enables the user to control his/her computer using visual and speech input. The package also includes training and test applications that enable users to become familiar with the system before they use it in real applications. Test applications enable the quantitative assessment of the performance of users when using our system. A number of volunteers tested our system and provided feedback related to the performance of the system. Both the feedback received and quantitative results prove the potential of using our system in real applications. The remainder of the paper is organized as follows: In section 2 we present a brief overview of the relevant bibliography and in section 3 we describe the design methodology of the proposed system. In section 4 we describe the functionality offered by the proposed system and in section 5 we present the familiarization and test applications. In sections 6 and 7 we present experimental results and concluding comments.

2 Literature Review

Toyama [15] describes a face-tracking algorithm that uses Incremental Focus of Attention. In this approach they perform tracking incrementally starting with a layer that just detects skin color and through an incremental approach they introduce more capabilities into the tracker. Motion information, facial geometrical constraints and information related to the appearance of specific facial features are eventually used in the tracking process. Based on this approach they achieve real time robust tracking of facial features and also determine the facial pose in each frame. Information related to the face position and pose is used for moving the cursor on the screen. Gorodnichy and Roth [5] describe a template matching based method for tracking the nose tip in image sequences captured by a web camera. Because the intensities around the nose tip are invariant to changes in facial pose they argue that the nose tip provides a suitable target for face tracking algorithms. In the final implementation cursor movements are controlled by nose movements, thus the user is able to perform mouse operations using

nose movements. Gorodnichy and Roth have used the system for several applications like drawing and gaming but they do not provide a quantitative evaluation of the proposed system.

Several commercial head movement-based HCI systems are available [1]. In most cases head tracking relies on special hardware such as infrared detectors and reflectors [13] or dedicated camera configurations [3],[10],[11]. Hands free non-invasive systems that only require standard web cameras are also available in the market [2].

Human Computer interaction based on speech has received considerable interest [12] since it provides a natural way to interact with a machine. However, under some circumstances speech-based HCI can be impractical since it requires quiet environments. In several occasions [14] speech recognition algorithms are combined with automatic lip-reading in order to increase the efficiency of speech HCI and make it more robust to speech recognition errors. A number of researchers describe multi-modal HCI systems that combine gesture input and speech. Such systems usually target specific applications involving control of large displays [6],[7]. With our system we aim to provide a generic speech-based HCI method rather than supporting a unique application.

3 Multi-modal Human Computer Interaction

We describe herein the face tracking algorithm and the speech processing techniques used as part of the multi modal HCI system.

3.1 Face Tracking

We have developed a face-tracking algorithm based on integral projections. An integral projection [8] is a one-dimensional pattern, whose elements are defined as the average of a given set of pixels along a specific direction. Integral projections represent two-dimensional structures in image regions using only two one-dimensional patterns, providing in that way a compact method for region representation. Since during the calculation of integral projections an averaging process takes place, spurious responses in the original image data are eliminated, resulting in a noise free region representation. In order to perform tracking based on this methodology, we calculate the horizontal and vertical integral projections of the image region to be tracked. Given a new image frame we find the best match between the reference projections and the ones representing image regions located within a predefined search area. The center of the region where the best match is obtained, defines the location of the region to be tracked in the current frame. This procedure is repeated on each new frame in an image sequence. The method described above formed the basis of the face-tracking algorithm employed in our system. The face tracker developed, tracks two facial regions - the eye and nose regions (see figure 2). The nose and eye regions are primarily used for estimating the vertical and horizontal face movement respectively.

During the tracker initialization process the face of the computer user is detected and the locations of the eyes and nose are established. In order to improve the robustness of the initialization algorithm and extend its applicability in cases where the lighting conditions are not ideal, the user of the system is requested to blink few times during

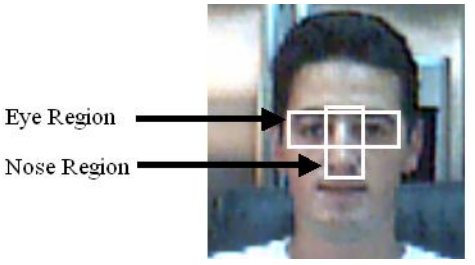


Fig. 2. The nose and eye regions

the initialization process. Once the eyes and nose are located, the vertical projection of the nose region and the horizontal projection of the eye region are calculated and used as the reference projections during tracking. During the tracking process the eye region and nose region are tracked in every frame, and once the position of the two regions is established the exact location of the eyes is determined by performing local search in the eye region.

In order to improve the robustness of the face tracker to variation in lighting, we employ intensity normalization so that global intensity differences between integral projections derived from successive frames are removed. Robustness to face rotation is achieved by estimating the rotation angle of a face in a frame so that the eye and nose regions are rotated prior to the calculation of the integral projections. Constraints related to the relative position of the nose and eye regions are employed in an attempt to improve robustness to occlusion and excessive 3D rotation. In this context deviations of the relative positions of the two regions that violate the statistical constraints pertaining to their relative positioning, are not allowed.

The results of a rigorous experimental evaluation [4] proved that the face tracking algorithm is capable of locating the eyes of subjects in image sequences with less than a pixel mean accuracy, despite the introduction of various destructors such as excessive rotation, occlusion, changes in lighting and changes in expression. Even in cases that the tracker fails to locate the eyes correctly, the system usually recovers in subsequent frames and re-assumes accurate eye-tracking.

3.2 Speech Processing

Instead of implementing our own speech recognition algorithms, we have employed the speech processing functionality offered by the Microsoft Speech Software Development Kit [9] that contains the Win32 Speech API (SAPI). SAPI provides libraries with dedicated functions for recording, synthesizing and recognizing speech signals. Our work in this area focuses on the development and testing of a suitable protocol to be used in conjunction with the head-based HCI system in order to allow computer users to achieve efficient hands-free control over a computer system. We have implemented two methods for using speech input: The Sound Click and the Voice Command Mode.

Sound Click: When the Sound Click mode is active, users activate mouse clicks just by creating a sound. In this mode we continuously record speech input and in the case

that an input signal stronger than the background noise is detected, a click is triggered. In this mode any sound of higher intensity than the background is enough to trigger a mouse click, hence this mode is not appropriate for noisy environments. The main advantages of the Sound Click mode are:

- Fast reaction time to user-initiated sounds enabling in that way real-time mouse click activation.
- When using the Sound-click it is not necessary to perform person-specific speech training.
- Any sound can be used for triggering a click, hence users may utilize sounds other than speech. (i.e users may tap on a surface in order to activate a click)

Voice Command: We have utilized speech recognition algorithms available in the SAPI in relation with an appropriate HCI protocol in order to add in our proposed hands-free HCI system, the ability to activate certain tasks by sound. Our ultimate aim is to improve the speed of activating frequently used HCI tasks. Our work in this area focuses on the specification of a suitable set of verbal instructions that can be recognized with high accuracy by the speech recognition algorithm. All verbal commands supported, have been separated into five groups according to the type of action they refer to. In order to

Table 1. Voice commands used in the system

Group	Command	Description
Mouse	Click	Perform Left Click
Mouse	Right-Click	Perform Right Click
Mouse	Drag	Hold left button down
Mouse	Drop	Release left button
Mouse	Scroll-Up	Scroll active window up
Mouse	Scroll-Down	Scroll active window down
Mouse	Stop	Stop face tracker
Mouse	Begin	Start face tracker
Move Cursor	Top	Move cursor to the top center of the screen
Move Cursor	Top-Right	Move cursor to the top right of the screen
Move Cursor	Top-Left	Move cursor to the top left of the screen
Move Cursor	Bottom	Move cursor to the bottom center of the screen
Move Cursor	Bottom-Right	Move cursor to the bottom right of the screen
Move Cursor	Bottom-Left	Move cursor to the bottom left of the screen
Move Cursor	Center	Move cursor to the center of the screen
Computer	Copy	Copy selected item
Computer	Paste	Paste
Computer	Enter	Press enter
Computer	Close	Close active window
Computer	Shut-down	Shut down computer
Computer	Sound	Enable Sound-Click mode
Open	Windows-Explorer	Run Windows Explorer
Open	Media-Player	Run Media Player
Open	Internet-Explorer	Run Internet Explorer
Open	Keyboard	Run the “On-Screen Keyboard” application

activate a specific command the user has to provide two keywords: The first keyword is used for specifying the group and the second one is used for specifying the exact command he/she wishes to activate. Both the groups and the commands in each group are user configurable - in table 1 we present the default selection of voice commands specified in the system.

While the Voice Command mode is active the system continuously records sounds. Once the system detects a sound with intensity higher than the background, it attempts to classify the sound to one of the group keywords. If none of the group keywords matches the sound, the system rejects the sound. In the case that a sound is recognized as a group-keyword, the system is expecting to receive a second sound corresponding to a sub-command of the activated group. Sounds recorded after a group keyword, are tested against the commands belonging to the corresponding group and if a match is detected the appropriate action is activated. In the case that a match is not detected, the input is rejected. The main reason for separating the commands in groups is to maximize the robustness of the speech recognizer by reducing the number of candidates to be recognized. Based on the proposed scheme a recorded word is classified only among the five keywords corresponding to each group. Once a correct group keyword is recognized the second word is classified based on the sub-commands for each group, instead of dealing with all system commands. In this way the probability of misclassifications is minimized and at the same time the tolerance of the system to background noise and microphone quality is maximized.

4 System Description

In this section we describe how various functions are implemented in the proposed non-invasive human computer interaction system. Those actions refer to system initialization, system training and simulation of click operations.

4.1 System Initialization

The first time that a user uses the system he/she is required to go through a training procedure so that the system learns about the visual and speech characteristics of the user. Although it is possible to use the system based on a generic training procedure, the overall system performance is enhanced when person-specific training is adopted. In order to train the face tracker a dedicated tool is used, where the user is requested to keep his/her face still and perform blink actions. Based on a frame-differencing algorithm the positions of the eyes and nose regions are determined and integral projections for those areas are computed. Once the projections are computed the face tracker is activated. The tracker initialization process requires approximately 10 seconds to be completed. A screen shot of the initialization tool is shown in figure 3.

When using voice input the user is advised to configure the microphone using a dedicated tool provided by the Microsoft Speech Software Development Kit [9]. Once the microphone is configured it is possible to detect input signals of higher intensity than the background noise. The microphone configuration process requires approximately 30 seconds to be completed. In the case that the system is used only in the

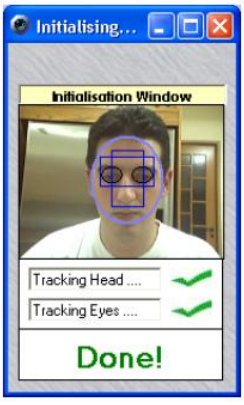


Fig. 3. System initialization window

Sound- Click mode it is not required to perform any person-specific training. When using the Voice Command mode the user is requested to read a sample text so that the system collects the necessary information required for speech recognition. The training of the speech recognizer is carried out using the Microsoft Speech Recognition Training Wizard, which is available in the Microsoft Speech Software Development Kit [9]. The speech recognition training process requires approximately 20 minutes to be completed. It is important to note that system initialization can be done in a hands free fashion (provided that the camera and microphone are already installed on the system). The tool used for visual initialization is activated during start up and once the face tracker is in operation the user can use head movements in order to initiate and complete the training for speech processing or activate his/her speech profile in the case of a returning user.

4.2 Activating Mouse Actions

In this section we describe how mouse operations are implemented in our system.

Moving the Cursor: The divergence of the face location from the initial location is translated in cursor movement speed, towards the direction of the movement. Based on this approach only minor face movements are required for initiating substantial cursor movement. The sensitivity of the cursor movement can be customized according to the abilities of different users. When the Voice Command mode is active, it is also possible to move the cursor to predetermined positions by recalling commands from the group "Move Cursor" (see table 1). The use of speech commands is useful for fast initial cursor positioning, followed by refinement of the cursor position based on head movements.

Mouse Click Actions: Three different methods for activating mouse click actions are provided:

– Idle Click: Clicks are activated by the stabilization of the cursor to a certain location for a time period longer than a pre-selected threshold (usually around one second). In this mode, users select beforehand the required click action to be activated when the cursor is stabilized. The predefined options include: left click, right click, double click, drag and drop and scroll.
– External Switch: Click actions are performed using an external switch attached to the system. In this mode the user directs the mouse to the required location and the appropriate click action is activated based on the external switch. The switch can be activated either by hand, foot or voice (when the Sound Click mode is enabled).
– Voice Command Click: Voice commands available in the group "Mouse" (see table 1) are used for activating the required mouse click actions.

Text Entry: Text entry is carried out by using the "On-Screen Keyboard" - a utility provided by the Microsoft Windows Operating System (see figure 4). Once the On-Screen Keyboard is activated it allows the user to move the cursor on any of the keys of the keyboard and use mouse click actions in order to press any key. As a result it is possible to use head movements and speech for inserting text or triggering any operation that is usually triggered from the keyboard.

Fig. 4. Screen-Shot of the "On Screen Keyboard"

5 Hands-Free Applications

Although the proposed multi-modal HCI system can be used for any task where the mouse and/or keyboard is currently used, we have developed dedicated computer applications that can be used by prospective users of the system for familiarization and system evaluation purposes. In this section we briefly describe the familiarization and test applications.

5.1 Familiarization Applications

We have developed three familiarization applications: A paint-tool application, a car racing game and a virtual piano. Screen shots of the familiarization applications are shown in figure 5. The paint-tool application allows the user to draw different trajectories on the screen using head movements. The aim of the paint-tool is to enable the user to get initial experience with head-based cursor control. The car racing game allows the

Fig. 5. Screen shots of the paint-tool (left), car racing (middle) and virtual piano (right) familiarization applications

user to use head movements in order to direct a car through a racing circuit. The aim is to complete the lap without collisions with the boundary. The car racing application allows the user to learn how to control the movement and the speed of the cursor through head movements. The virtual piano application allows the user to learn how to move the cursor to a specific position on the screen (a piano key in this case) and activate mouse-click operations in order to play piano. It is highly recommended that new users of the system use the familiarization programs before they use the system for real life HCI applications.

5.2 Test Applications

Test applications are used as a test bench for obtaining quantitative measurements related to the performance of the users of the system. Screen shots of the three test applications used in our experiments are shown in figures 6, 7 and 8. In the following paragraphs we describe the purpose and the functionality of each test application.

Click Test: The user is presented with four squares on the screen. At any time one of those squares is blinking and the user is requested to direct the cursor and click on the blinking square. This process is repeated several times and at the end of the experiment the average time required to direct the cursor and click on the correct square is quoted. The aim of this test application is to asses the ability of a user to direct the cursor to a specific screen location and perform a mouse click operation.

Draw Test: The user is presented with different shapes on the screen (square, triangle and circle) and he/she is asked to move the cursor on the periphery of each shape. The divergence between the actual shape periphery and the periphery drawn by the user is quoted and used for assessing the ability of the user to move the cursor on a predefined trajectory.

Typing Test: The user is presented with a word and he/she is asked to type the word presented. This procedure is repeated for a number of different randomly selected words. The average time required for typing a correct character is quoted and used for assessing the ability of the user to type text.

Fig. 6. Screen shot of the click-test application

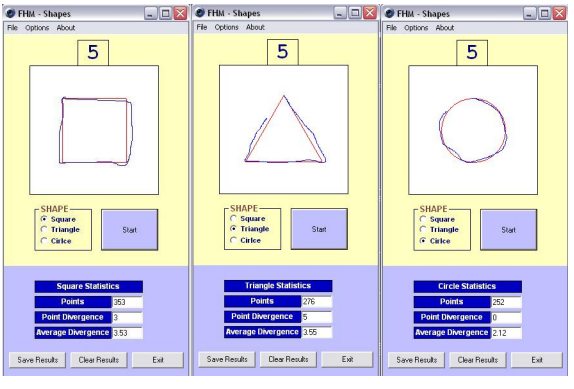


Fig. 7. Screen shots of the draw-test application

Fig. 8. Screen shot of the typing-test application

6 System Evaluation

The test applications presented in the previous section were used for assessing the usefulness of the proposed system. In this section we describe the experiments carried out and present the results.

6.1 Experimental Procedure

Twenty volunteers tested our system in order to obtain quantitative performance evaluation results. The test procedure included the following steps:

Familiarization Stage: Each volunteer is instructed how to use the hands-free computing system and he/she is allowed to get familiar with the system by using the familiarization applications. On average the duration of the familiarization stage was about 15 minutes.

Benchmark Performance: Each volunteer was requested to complete all test applications using a conventional mouse and a typical touch pad of a portable personal computer. The performance of each volunteer is assessed on the following tests:

- Click test: The average time required for five clicks is recorded.
- Draw test: The subject is asked to draw a square, a triangle and a circle and the average discrepancy between the actual and the drawn shape is quoted.
- Type test: The user is asked to type five randomly selected 3-letter words and the average time for typing a correct letter is recorded (For this test the "On Screen Keyboard" was used for text input).

The results obtained from this exercise define the benchmark performance for the test applications.

System Performance: The performance of the system is assessed by allowing the volunteers to complete the same tests as the ones carried out during the definition of the benchmark performance. However, in this case users direct the mouse using head movements and perform click actions using the following options.

- Idle-Clicks: Mouse clicks are activated using the idle time click method.
- Eternal Switch: Mouse clicks are activated using a hand-operated external switch.
- Sound-Click: Mouse clicks are activated using the sound-click method.
- Voice-Command: Mouse clicks are triggered using voice commands.

The 20 volunteers who tested the system were separated into two groups according to their prior expertise in using the hands-free computing system. Group A contains subjects with more than five hours prior experience in using the hands free system. Subjects from group B used the system only as part of the familiarization stage (for about 15 minutes). All tests were carried out in standard office environments - no precautions for setting up lighting conditions or for minimizing background noise were enforced.

6.2 Results – Discussion

The results of the tests are summarized in table 2. Based on the results the following conclusions are derived:

Click Test: In all occasions the results obtained by using a conventional mouse are better. When the hands free system is combined with an external switch for performing click actions the performance of the system is comparable with the performance achieved when using a touch pad. In the case that the Sound-Click mode is used, the performance of the users compares well with the performance achieved when using a mouse. When idle-clicks are used, the performance of the users decreases. The additional delay introduced in this case is mainly due to the requirement for stabilizing the cursor for some time (one second according to the default setting) in order to activate a click action.

Table 2. Quantitative Evaluation Results. (For the click and and typing tests numerical figures show the number of seconds required per click. For the draw test numerical figures indicate the divergence between actual and drawn shape outlines, in pixels.).

Test	Method	Group A	Group B
Click Test	Mouse	0.76	0.86
Click Test	Touch Pad	1.45	2.18
Click Test	Idle-Clicks	3.58	4.84
Click Test	External switch	1.41	2.50
Click Test	Sound-click	0.98	1.68
Click Test	Voice Command	2.10	2.38
Draw Test	Mouse	2.62	2.05
Draw Test	Touch Pad	3.01	4.01
Draw Test	Head Movements	3.07	5.93
Typing Test	Mouse	0.89	0.86
Typing Test	Touch Pad	1.73	3.41
Typing Test	Idle-Clicks	4.39	5.99
Typing Test	External switch	2.37	3.70
Typing Test	Sound-click	2.33	3.84
Typing Test	Voice Command	3.19	5.53

Draw Test: For experienced users of the system (Group A) the performance achieved using the free hand mouse is comparable with the performance achieved when using a touch pad. Subjects from group B (inexperienced users) produced an inferior performance when using the hands free system. The main reason is the reduced ability to control precisely cursor movements due to the limited prior exposure to the system.

Typing Test: In this test the use of mouse or touch pad for typing text is significantly superior to the performance of users using the hands-free system, indicating that the proposed system is not the best alternative for typing applications. However, the

performance obtained when using the hands-free system in conjunction with the external switch or Sound-click, is once again comparable to the performance obtained with the touch pad. The main reason for the inferior performance obtained when using the hands free system, is the small size of the keys on the "On Screen Keyboard" that requires precise and well-controlled cursor movements. The ability to precisely move the cursor requires extensive training. Instead of using the "On Screen Keyboard", provided by the Windows operating system, it is possible to use dedicated virtual keyboards with large buttons in order to improve the typing performance achieved when using the hands-free computing system.

User Expertise: The abilities of users to use the hands free system increase significantly with increased practice. Based on the results we can conclude that subjects with increased prior experience in using the hands-free system (from group A) can perform all usual HCI tasks efficiently. It is expected that with increased exposure to the system, users will be able to achieve even better performance.

External Switch: The introduction of an external switch that can be activated either by foot or hand or voice enhances the performance of the system.

Voice Command: When using the Voice Command mode, additional delays are introduced, due to the processing time required for performing the speech recognition task. However, when the task we wish to perform is contained in the Speech Command menu (see table 1), then a speed up in the overall task completion time can be achieved. For example the time required to verbally activate an application among the ones listed in the "Open" group menu (see table 1), is far less than in the case of using the rest of the methods.

7 Conclusions

We presented a prototype multi-modal hands-free HCI system that relies on head movements and speech input. The proposed system caters for common HCI tasks such as mouse movement, click actions and text entry (in conjunction with the "On Screen Keyboard"). Based on the quantitative results presented, head based HCI cannot be regarded as a substitute for the conventional mouse, since the speed and accuracy of performing most HCI tasks is below the standards achieved when using a conventional mouse. However, in most cases the performance of the proposed system is comparable to the performance obtained when using touch pads of portable computer systems. Even though the accuracy and speed of accomplishing various HCI tasks with a touch pad is less than in the case of using a conventional mouse, a significant number of computer users use regularly touch pads. We are convinced that computer users will also find the proposed hands free computing approach useful.

The proposed system does not require extensive person-specific training, since the system adapts and learns the visual characteristics of the features to be tracked, during the initialization phase. The only case that extensive person-specific training is required is when the "Voice Command" mode is used. The training procedure in those cases

requires about 20 minutes to be completed. The speech recognition training procedure is carried out once for each user of the system.

The proposed system is particularly useful for paraplegics with limited (or without) hand mobility. Such users are able to use a computer system based only on head movements and speech input. During the system development phase we have provided the system to members of the Cyprus Paraplegics Organization, who tested the system and provided valuable feedback related to the overall system operation and performance. Currently a number of paraplegic computer users are using the hands-free system described in this paper.

An important feature of the system is the provision of alternative methods for performing a task, so that at any time the user can choose the most appropriate way to perform an action. For example if the user wishes to run the Internet Explorer, he/she has the ability to perform the action using only head movements or by using speech commands or by using a combination of the two input media (i.e move the cursor to the appropriate icon using head movements and run the application by using sound clicks).

In the future we plan to upgrade the Voice Command mode in order to allow text entering based on speech input. Also we plan to stage a large-scale evaluation test is order to obtain concrete conclusions related to the performance of the proposed system. Since the hands-free system is primarily directed towards paraplegics, we plan to include evaluation results from paraplegics in our quantitative evaluation results.

Acknowledgements

The work described in this paper was supported by the Cyprus Research Promotion Foundation. We are grateful to members of the Cyprus Paraplegics Organization for their valuable feedback and suggestions.

References

1. Assistive Technology Solutions (Last Accessed: July 2007), http://www.abilityhub.com/mouse/
2. CameraMouse: Hands Free Computer Mouse (Last Accessed: July 2007), http://cameramouse.com/
3. EyeTech Digital Systems-Eye Tracking Device (Last Accessed: July 2007), http://www.eyetechds.com/
4. Frangeskides, F., Lanitis, A.: A Hands-Free Non-Invasive Human Computer Interaction System. In: Proceedings of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (2005)
5. Gorodnichy, D.O., Roth, G.: Nouse 'Use Your Nose as a Mouse' - Perceptual Vision Technology for Hands-Free Games and Interfaces. Image and Vision Computing 22, 931–942 (2004)
6. Kettebekov, S., Sharma, R.: Toward Natural Gesture/Speech Control of a Large Display. In: Nigay, L., Little, M.R. (eds.) EHCI 2001. LNCS, vol. 2254, pp. 221–234. Springer, Heidelberg (2001)
7. Krahnstoever, N., Kettebekov, S., Yeasin, M., Sharma, R.: A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays. In: ICMI 2002. Proceedings of Fourth Intl. Conference on Multimodal Interfaces (2002)

8. Mateos, G.G.: Refining Face Tracking With Integral Projections. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 360–368. Springer, Heidelberg (2003)
9. Microsoft Speech - Speech SDK5.1 For Windows Applications (Last Accessed: July 2007), http://www.microsoft.com/speech/download/sdk51/
10. Mouse Vision Assistive Technologies (Last Accessed: July 2007), http://mousevision.com/
11. Origin Instruments Corporation (Last Accessed: July 2007), http://www.orin.com/3dtrack/
12. O'Shaughnessy, D.: Interacting with computers by voice: automatic speech recognition and synthesis. IEEE Proceedings 91, 1272–1305 (2003)
13. Smart-Nav: Hands Free Mouse AT Assistive Technology (Last Accessed: July 2007), http://www.naturalpoint.com/smartnav/
14. Potamianos, G., Neti, C., Gravier, G., Garg, A., Senior, A.W.: Recent advances in the automatic recognition of audiovisual speech. IEEE Proceedings 91, 1306–1326 (2003)
15. Toyama, K.: Look, Ma - No Hands! - Hands Free Cursor Control with Real Time 3D Face Tracking. In: Proceedings Of Workshop on Perceptual User Interfaces, pp. 49–54 (1998)

The ViskiMap Toolkit: Extending Mediawiki with Topic Maps

Cleo Espiritu, Eleni Stroulia, and Tapanee Tirapat

Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada
{cleo,stroulia,tapanee}@cs.ualberta.ca

Abstract. In this paper, we present our ViskiMap systems, ENWiC (EduNuggets Wiki Crawler) and Annoki (Annotation wiki), for intelligent visualization of Wikis. In recent years, e-Learning has emerged as an appealing extension to traditional teaching. To some extent, the appeal of e-Learning derives from the great potential of information and knowledge sharing on the web, which has become a de-facto library to be used by students and instructors for educational purposes. Wiki's collaborative authoring nature makes it a very attractive tool to use for e-Learning purposes. Unfortunately, the web's text-based navigational structure becomes insufficient as the Wiki grows in size, and this backlash can hinder students from taking full advantage of the information available. The objective behind ViskiMap is to provide students with an intelligent interface for navigating Wikis and other similar large-scale websites. ViskiMap makes use of graphic organizers to visualize the relationships between content pages, so that students can easily get an understanding of the content elements and their relations, as they navigate through the Wiki pages. We describe ViskiMap's automated visualization process, and its user interfaces for students to view and navigate the Wiki in a meaningful manner, and for instructors to further enhance the visualization. We also discuss our usability study for evaluating the effectiveness of ENWiC as a Wiki Interface.

Keywords: Novel E-learning interfaces and interactions, Web-based education software, Intelligent information and knowledge management systems, Intelligent visualization tools.

1 Introduction

Wikis and blogs are becoming increasingly popular means of on-line communication, collaboration and information sharing. As of August 13th 2007, the English Wikipedia [24] contains close to 2M articles and the numbers increase daily. Furthermore, an abundance of domain-specific wikis exist on a variety of topics, such as, for example, computing science [19], [16], image sharing [18], etc.

The collaborative nature of information publishing in Wikis makes them an attractive tool for e-Learning purposes. Wikis are an ideal medium for instructors and students to interact. It is easy to add and modify Wiki content and to establish links between Wiki pages. This ease of use strongly encourages users to contribute content, giving the Wiki an incredible wealth of information. However, this wealth does come at a cost: the more

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 420–438, 2008.

pages a Wiki has, the more difficult it becomes to navigate through it. It is common for a Wiki page to contain a sizable number of links in its body text. Users have to read through the text to locate these links and to guess what the next appropriate pages might be. It is also easy for users to feel "lost" after having gone through a number of links, since the standard wiki interface does not provide any navigation context. Furthermore, the nature of the relationship between a page and its neighbors is not always intuitive, since wikis do not impose any requirements on what links are appropriate and do not support any metadata for link annotation. To alleviate this problem, some Wikis provide special purpose pages, such as roadmaps, categories, and lists of back links. However, it is not immediately apparent to the users how to utilize these pages; novice wiki users, especially, may not even be aware of the purpose and usage of such pages. Even these special-purpose pages are typically overwhelmingly long lists of text links.

The objective of this work is to provide a more intuitive user interface to wikis, in order to take advantage of the large information repositories that they provide. We are especially interested in supporting student usage of wikis offering technical content. Students accessing these wikis need an easy-to-learn and easy-to-use interface that will enable them to effectively locate information relevant to their tasks and, in the process, to achieve a deeper understanding of the concepts presented in the various wiki pages.

Our ViskiMap (*Vis*ualizing Wi*ki*s as Topic *Map*s) toolkit, which consists of ENWiC (EduNugget Wiki Crawler) and Annoki (Annotation Wiki), builds on our previous work on the EduNuggets framework [14], whose aim was to organize educational documents into topic maps. A topic map models the semantics of an application domain in terms of topics, associations and their instances. Thus, it imposes a structure to the domain data, which can also provide the basis for a graphical visualization. To ease the topic-map creation process, users can either use the ENWiC web crawler to generate a topic map from wiki pages, or build their own topic maps through the Annoki wiki. Further refinements can be made by authoritative experts – such as instructors or wiki moderators – to add concepts and structure to the topic map. Since graphic organizers have been shown to be useful in assisting students in learning concepts and identifying relationships between concepts [1],[12], we visualize the Wiki topic map as a graph to help students navigate the Wiki. By defining different types of associations in the topic map, different types of graphic organizers can be created to support each type of cognitive activity in Bloom's taxonomy [7]. This will help students achieve different levels of understanding of the content faster than using only indices and text of the Wiki.

The rest if the paper is organized as follows: Section 2 discusses related research; Section 3 describes the goals behind ENWiC and Annoki, and gives an overview of the two components; Section 4 reviews them in detail; Section 5 discusses our empirical evaluation of ENWiC; and finally, Section 6 concludes with a summary of our findings.

2 Related Research

2.1 Wikis

The Platypus [4] and SHAWN [2] tools use semantic-web technologies to describe the ontology of a Wiki. Both utilize RDF (Resource Description Framework) metadata to

structure and organize the Wiki content. In the Platypus Wiki, users are provided with a standard Wiki editor and a companion metadata editor for each Wiki page. Users can set up a site link table which maps a keyword to a URL; when another user writes a Wiki page, the tool will automatically look for keywords from the site link table and covert them to links to the corresponding URLs. When viewing a page, the metadata is shown in a side panel to help with navigation. In SHAWN, users can edit both the metadata and the content of a page in the same editor. SHAWN automatically displays forward links and pages with relating metadata based on the triplets entered by the author(s) of the page.

Semantic Mediawiki [21] extends the functionalities of Mediawiki [17], the software of Wikipedia [24], to enable users to add more semantics to a wiki page and provide better search results.

The slide2wiki tool [11] allows educators to generate wiki pages from their presentation slides. Students and teachers can then annotate these wiki pages with their own comments.

Wang and Turner [15] tailored a wiki application for online course delivery. The application access controls allow teachers to be in control of what pages can be viewed and/or modified by students. Students also have access control over pages they created to prevent other students from modifying their work.

2.2 Visualization of Wikis and Other Online Resources

Touchgraph [23] released a Wiki browser for the Meatball Wiki and the Emacs Wiki to demonstrate their toolkit. It uses the link database of a Wiki to create the graph. The downside to this browser is that it only works with Wikis that have a link database, and there is no easy way to edit the graph itself if a user wishes to do so, unless the user manipulates the links on the actual wiki pages.

TM4L (Topic Map For Learning) [6] provides an editor and a viewer for topic maps, designed for e-Learning purposes. It supports pre-defined associations, relationships and topic types to allow for easy creation of learning content. It also supports user-defined association types as well. The Viewer visualizes the topic map in a graph or a tree, but it does not provide a browser to let user directly access the URIs that a topic node may contain.

Smith [13] developed a set of visualization tools for Usernet. Usernet can grow to contain a very large number of messages, and like a Wiki, a typical Usernet client just does not have an adequate organization and navigation structure to effectively convey the ontology of the messages. Smith uses a number of information types – such as the history of a thread and the participation history of each user – to create a variety of views to represent the threads in a newsgroup.

Canas and his colleagues [5] described how concept maps, a type of graphic organizer, can be used to visualize ‘knowledge’ to assist users in organizing information and searching information on the web. Their CmapTools represents knowledge as concepts and propositions in a map. CmapTools utilizes the concept maps to enhance web search; a user can simply click on a concept node, and search queries are generated based on relating concepts and authority concepts. After executing the search, the results are ranked based on the distance matrices from the map.

Like CmapTools, VisSearch [10] uses a graph to enrich search results. Users create a search graph that contains query nodes and associations between them. The search graph can be reused; users can select multiple nodes, which will combine the existing queries to form a new query. Users can also associate relevant bookmarks to a query node as a means of organizing useful links. VisSearch also takes advantage of search results from multiple users to recommend potentially useful sites. A VisSearch server takes search results from users with similar interests and analyzes the data so that when another user searches for similar information, it can determine what results may be interesting to the user.

3 System Requirements

The primary objective of the ViskiMap toolkit is to provide an intuitive interface for users to effectively navigate the Wiki structure so that they can find information relevant to their tasks. This is achieved by providing users with an interactive graphical interface that visualizes the Wiki as a topic map. There are two intuitions behind the design of ViskiMap. First, a topic map provides an overall view of the wiki organization structure, thus enabling users to maintain a mental model of their browsing history and current context. The topic map can be visualized at different levels of detail, so that users can control the amount of information presented to them. Second, graphic organizers are used to represent distinct types of substructures, easily recognizable by the users who may choose to focus on them when they correspond to the types of relations they are interested in. At the same time, as with the standard wiki interface, users can always jump to a specific topic using a search function, instead of browsing through the graph.

In addition to this primary functional requirement, the ViskiMap toolkit was designed with the following quality requirements in mind:

1) *Automatic topic-map discovery:* The topic-map representation of the Wiki is generated automatically, based on the wiki-pages hyperlink structure. Each page is viewed as a topic, while each link between two pages is interpreted as an association.
2) *Easy authoring of domain-dependent semantics*: Users are able to add domain-dependent semantics to generate graphic organizers that are not discovered from the wiki-page hyperlink structure.

To satisfy the first requirement, we took into consideration that users may or may not have control over the wiki software. Some users may wish to use external sources, while some may wish to install their own database and wiki software. ENWiC is designed to target the former group; its web crawler analyzes generic HTML tags to discover semantics and links to other pages. Thus, ENWiC also achieves *implementation independence*, as it will work with any wiki implementations that serve HTML pages. If the users can adopt a specific wiki implementation, they can use the Annoki wiki (based on MediaWiki 1.5.5 [17]), which allows topic maps to be generated directly from the database. This allows Annoki to produce *instant up-to-date topic maps*.

The second requirement is achieved by offering instructors an authoring environment that allows them to modify the automatically-generated topic map to enrich its

content. ENWiC provides an administrator tool equipped with wizards, which allow instructors to add extra graphic organizers with ease. In Annoki, instructors can simply modify the wiki pages to make new topics and associations. They can also add tags to the wiki pages to add hierarchical structures to the pages.

3.1 Bloom's Taxonomy Graphic Organizers

The design of the ViskiMap toolkit is largely inspired by the use of graphic organizers in support of learning. Graphic organizers such as spider maps, hierarchical diagrams and charts can persuade students to engage in a different kind of thinking instead of just comprehending from text [9]. Graphic organizers let student see the 'bigger picture' instead of just a part of the content. This addresses the problem with a standard Wiki, where the action of clicking through discrete pages only exposes parts of the Wiki to the users.

Bloom's taxonomy [3] describes six different levels of cognitive understanding that a learner can gain on a subject. Mastery at the *knowledge* level implies a learner's ability to recognize and recall information on the subject. Mastery at the *comprehension* level is accomplished when the subject has grasped the domain's semantics and can interpret and analyze facts to infer conclusions and make predictions based on them. At the *application* level, the learner is able to apply the knowledge in new contexts. A learner who has mastered a domain at the *analysis* level can see patterns and implicit organization in the domain structure and at the *synthesis* level he is able to generalize form the give fact and apply knowledge from many areas to construct new information. Finally, at the *evaluation* level, the learner can assess presentation and theories about the domain.

Donelan [7] relates a taxonomy of graphic organizers to Bloom's levels of understanding.

1) At the knowledge level, *spider maps* and *linear strings* support the learners' ability to remember basic facts on the subject.
2) At the comprehension level, *hierarchy diagrams* enable them to understand the semantic organization of the domain concepts.
3) At the application level, *flow charts* guide learners to apply known facts to solve problems.
4) At the analysis level, *fishbone* and *concept map*s reflect the learners' ability to explain facts.
5) *Idea map* – at the Synthesis level - reflects the ability to combine known facts to form solutions for problems.
6) *Double-cell diagrams* and *comparison matrices* - at the Evaluation level - reflect the ability to rate concepts and information.

ENWiC supports all these Bloom's graphic organizers. The underlying intuition is that when these organizers are used to present the domain knowledge visually, the learners can pick up the visual clues and accomplish a deeper level of mastery faster. The ENWiC crawler automatically builds different types of associations among topics and the ENWiC viewer visually reflects these relations. Additional graphical organizers can be created using the ENWiC administration tool.

Annoki concentrates more on letting users build hierarchy diagrams and spider maps through the use of special-purpose tags. Users can tag pages with a keyword representing the central idea common between the pages. They can also tag pages with property/value pairs to introduce simple hierarchical structures to the pages.

4 The ViskiMap System Architecture

Fig. 1 summarizes the components of ViskiMap. The three core entities in the systems are:

1) *The resource domain*, which consists of all the content written to the wiki(s) by authoritative experts, as well as the wiki software;
2) *The semantic representation of the domain*, which is the topic map generated based on content from the resource domain; and
3) *The client interface*, which is the interactive visualization of the topic map.

Users are presented with parallel interfaces; they can view, add and edit content using the native wiki interface from the resource domain; they can also navigate the content from the visualization interface.

4.1 ENWiC

As shown in Fig. 1, the ENWiC subsystem consists of three major system components: the *crawler*, the *administrator,* and the *viewer*. The resource domain consists of a collection of wiki pages that the users wish to explore. The Crawler, given a starting page URL, visits all wiki pages in the domain through links, and produces the topic map representation of the domain based on the data it discovers in the pages. The Administrator enables authoritative content providers to add and edit topics and associations to the crawler-produced topic maps. Finally, using the topic maps produced by the crawler and/or the administrator component, the Viewer provides a graphical interface to the original wiki content. The design of the Administrator is targeted for instructors who wish to provide students with extra semantics to the topic maps produced by the crawler, while the Viewer is designed for students to study the available Wiki domains and navigate through the topics.

The Crawler, given the URL of a starting HTML page, traverses all the Wiki pages in a breath-first search manner. It discovers new pages by extracting the hyperlinks included in each examined page. Each traversed page is stored in the ENWiC database as a topic under a domain specified by the user to correspond to the web site. Links between pages are marked with an 'instance of' relationship; thus, the path that the crawler has traversed to discover a page starting from the initial URL can be found by examining the linear sequence of the 'instance of' relationships. Since there are multiple alternate paths to a page that the crawler may discover as it examines more pages, the crawler marks these alternate links as associations. They are displayed in different colors in the Viewer so the users can identify them as alternate paths.

Each page usually contains several segments, separated from each other with boldface headings. The crawler attempts to classify them so that they can be visually

presented in a more organized manner. This is accomplished by extracting the text encapsulated in HTML header tags on a page. Each extracted header is made into a topic, and links that fall under that header become associated with that header's topic in the topic map. In the application, the header topics have the text 'SECTION' in their names to indicate that they only account for a section of a page, and not the full page. In the graph, their nodes are colored differently from the regular page nodes.

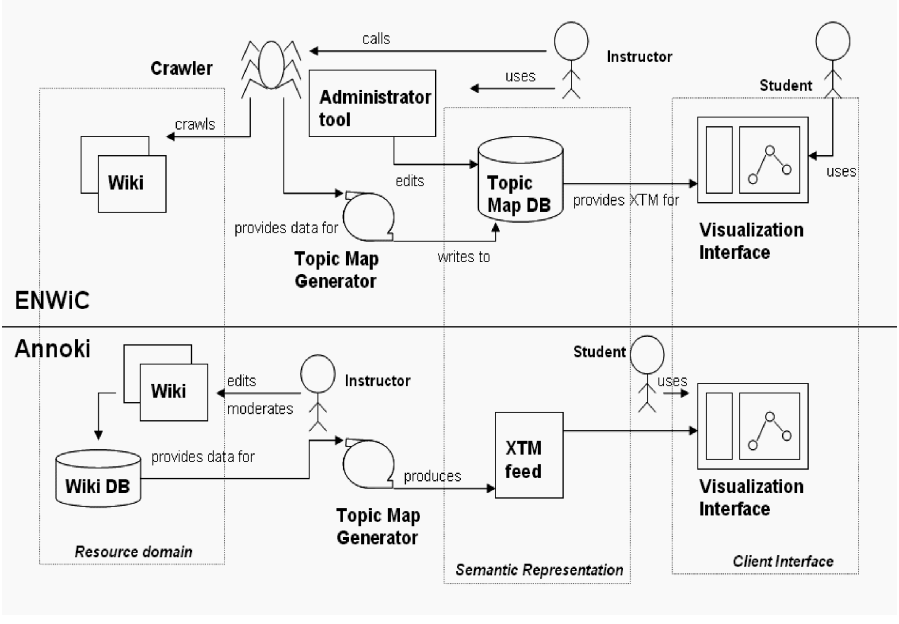


Fig. 1. Overview of the ViskiMap tools: ENWiC (top) and Annoki (bottom). The three core entities of the systems (the resource domain, the semantic representation of the domain, and the client interface) are outlined with dotted rectangles.

For each page discovered, the crawler uses the title of the page, the text of the link leading to the page, and the header text to create the topic's name and index terms. The crawler also ranks each page according to the number of links going to the page (link-in rank), and number of links going out of the page (link-out rank). The rank information is also visualized by the Viewer in order to help students identify which pages may be more useful; if a page is linked by many other pages then that page may contain some very useful information, and if a page has many links going out then that page may be a good index of relating pages.

The Viewer consists of two main components:

1) A *visualization panel* that displays a graph to represent the Wiki domain.
2) An *information panel* that provides: a master list of topics in the domain; information on the selected topic in the graph; a browser that displays the selected topic's Wiki page; a search function; and a history of recently visited topics.

The Viewer is implemented based on Touchgraph [23], which takes XML topic maps (XTM) [22] as input, and outputs the topic map as a graph. When users select the domain they wish to explore in the Viewer, the Viewer exports the corresponding topic map from the database in XTM format, which is then used to produce the visualization graph. Topic nodes that represent pages are drawn as a rectangle while nodes that represent sections are drawn as a round rectangle. Page nodes are colored based on rank; a gradient paint of purple and blue is used to represent the link-in rank and the link-out rank respectively. A deeper color indicates a higher rank. Fig. 2 shows the Pattern Roadmap page of the Portland Pattern Repository's Wiki [20] as seen with the ENWiC viewer.

Edges are colored and annotated based on the type of association. The user can select which type of edges to show to single out diagrams of interest, as shown in Fig. 3.

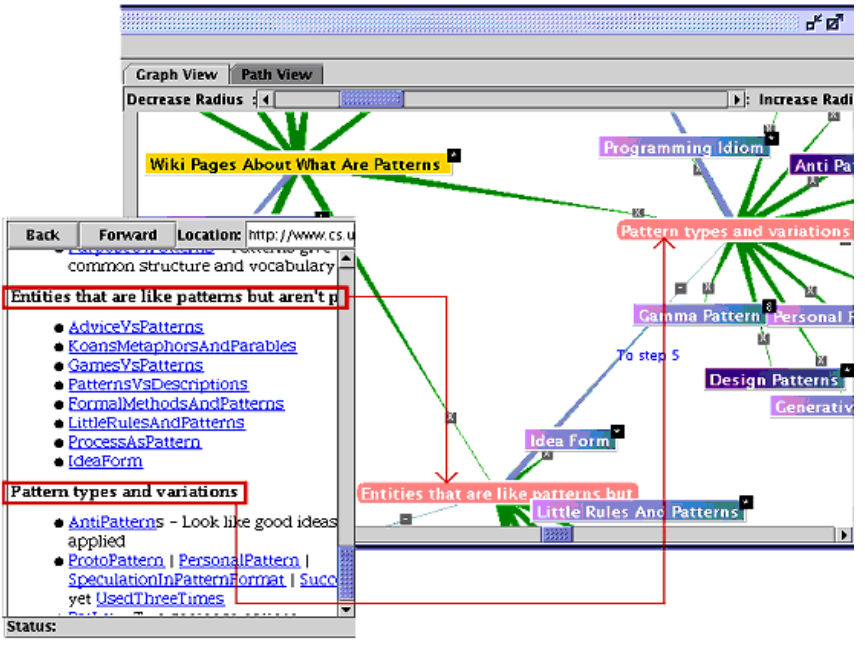


Fig. 2. The ENWiC Viewer. Blue/purple rectangle nodes represent page nodes while pink rounded-rectangle nodes represent sections of a page. The yellow node is the currently selected node.

The Viewer is equipped with a search function so that users can directly look for topics of interest and jump to the corresponding node in the graph immediately. This is useful when the domain has a large number of topics and users may not be able to locate specific topics in the graph readily. The search function allows them to isolate the desired topic node immediately and view the relationships it shares with other topic nodes. The Viewer also keeps a history of visited topic nodes so that users can

backtrack easily. The history is also visualized in the History View as a graph that will show any diverging paths that the user has taken.

The Administrator allows users to enter domain-dependent information as addendum to the domain-independent information discovered by the crawler. The user can modify any topic or association using the application.

Bloom's diagrams can be built using the Graphic Organizer Wizards provided by the application. The wizards allow users to easily add more semantics and relations to the crawler-produced topics. Each type of diagram created using the wizards will be color-coded in the Viewer so it would be easy for users of the Viewer to identify the types.

The Administrator component can also act as a simple graphic organizer builder. Users can start a new domain and add topics and associations manually or with the wizards. This is useful if the user just wanted to use a few pages to create a meaningful diagram, instead of using the entire wiki discovered by the crawler.

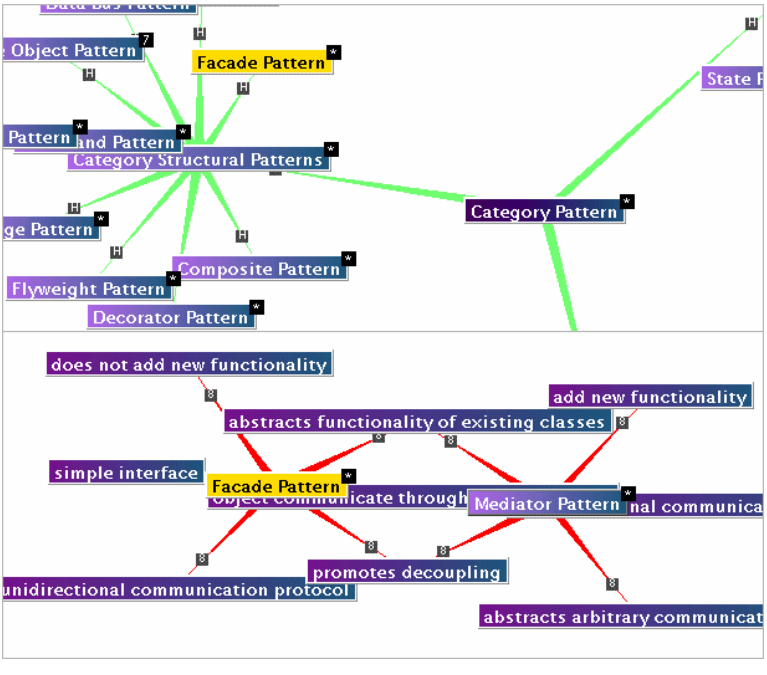


Fig. 3. Isolating the hierarchy diagram (top) and double cell diagram (bottom) using filters help users see different information about Facade pattern

4.2 Annoki

While the ENWiC Crawler can obtain a good topic map representation of the domain, having control over how data is organized in the wiki resource domain can greatly

improve the quality and accuracy of the topic map. Given a specific wiki implementation, we can incorporate additional semantics to the wiki to generate better topic maps. We also have the ability to generate the topic map directly from the database, which can reduce the time required to create a topic map. This is particularly important when we are dealing with large resource domains, which drastically increases the processing time needed for a crawler to transverse through all the pages. Exporting the topic map directly from the database also ensures every page in the wiki would be accounted for, as the ENWiC crawler is dependent upon the need for the wiki pages to be reachable through links from the starting page. We can also obtain user information and edit histories which allow us to incorporate user activities to the topic map.

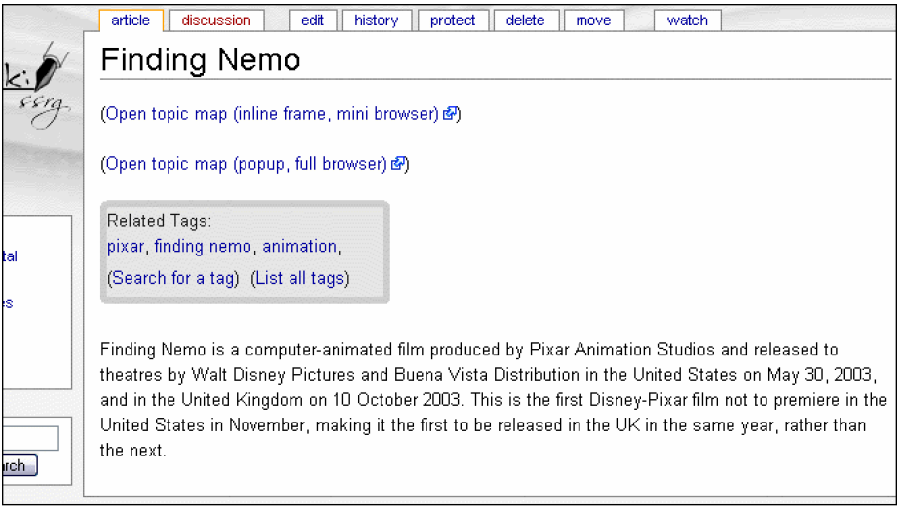


Fig. 4. A page from the Annoki Wiki

Annoki is designed with the same principles and requirements as ENWiC, except it drops the implementation independence requirement in exchange for the ability to produce a more accurate and up-to-date semantic representation of the domain.

As shown in Fig. 1, the resource domain of Annoki consists of the Annoki wiki and the data users wrote to the wiki. A Topic Map Generator queries the wiki database to produce the topic map representation of the domain, which is broadcasted as a feed in XTM format [22]. Similar to ENWiC, the client interface takes the topic map and produces the visualization graphs.

The Annoki wiki is based on Mediawiki [17]. Along with the features and functionalities offered in Mediawiki, users are presented with additional forms and menus to add and modify tags on the page. They can also launch the visualization interface from any page to see the topic map that is centered on that particular page. Fig. 4 shows a page from the Annoki wiki.

To tag a page, users can either enter keywords, or property/value pairs (written as 'property:value'). Keywords allow users to associate a set of pages through common

ideas and concepts. If they wish to better organize these ideas and concepts, they can use property/value pairs to impose a hierarchical structure on the semantics. Fig. 5 demonstrates the use of keywords and property/value pairs. Users can search wiki pages by tags, or look for property/value pairs by entering search terms for property, which will return all property/value pairs with that property.

Fig. 5. Associating 'Dory' and 'Nemo' through tags. They are linked by the common keyword 'finding nemo'. A hierarchy also links them through the root property 'fish', as Nemo is tagged with fish:clownfish while Dory is tagged as a fish:regal tang, indicating that both belong to the family of fish.

The Topic Map Generator is part of the Annoki wiki software. Users can obtain the topic map of the wiki in XTM format via the URL http://annokiURL/index.php?action=xml. Accessing this URL triggers the Generator to export entries from the relevant tables in the wiki database into a topic map. The steps taken by the Generator are as follows:

1) Obtain all tags in the wiki. For each tag:
   a. if the tag is a keyword, create a topic with the keyword as the topic name.
   b. if the tag is a property/value pair, create a topic for the property and a topic for the value. Create an association between the property and the value.
2) Obtain all pages from the wiki and create a topic for each page, with the page title as the topic name. For each page, query for relating tags and create an association between each relating tag and the page topic.
3) Obtain all users from the wiki and create a topic for each user, with the user name as the topic name. For each user, check user edit history. For each page the user has edited, create an association between the user and the page.
4) For every page-to-page link found, create an association between the two corresponding page topics.
5) Fig. 6 shows a segment of the XTM produced by the Generator.

```
<topic id="pg1283">
<baseName>
<baseNameString>PAGE: Marvin</baseNameString>

<variant><variantName>
<resourceData id="Nugget">text/html,
http://www.annokiURL/index.php/Marvin,
</resourceData></variantName></variant>

<variant><variantName>
<resourceData id="RevisionAnnotationCounters">RevAnn: 5:
3</resourceData>
</variantName></variant>

<variant><variantName>
<resourceData id="HitCounter">Hits: 71</resourceData>
</variantName></variant>

</baseName>
</topic>

<topic id="usr2">
<baseName>
<baseNameString>USER: Mary</baseNameString>
<variant><variantName>
<resourceData id="Nugget">text/html,
http://www.annokiURL/index.php/User:Mary,
</resourceData></variantName></variant>
</baseName>
</topic>

<association id="a48">
<instanceOf><topicRef xlink:href="#at-revby"/></instanceOf>
<member><roleSpec><topicRef
xlink:href="#pg1283"/></roleSpec></member>
<member><roleSpec><topicRef xlink:href="#usr2"/></roleSpec></member>
</association>
```

Fig. 6. Segment of the XTM file produced by the topic map generator in Annoki, showing a topic of a page (Marvin), a topic for a user (Mary), and an association that indicates Mary made revisions to Marvin

The Visualization Interface of Annoki is similar to the ENWiC viewer. It consists of an information panel on the left and a visualization panel on the right. With the Annoki wiki, we have the advantage of giving users the option to launch the visualization as an applet directly from a wiki page. The topic map centers on the page where the visualization is launched from, which eliminates the need for users to locate the page node on the graph manually. Users can choose to launch a smaller version of the visualization interface as a frame on the wiki page to view both interfaces simultaneously, or launch the full version of the visualization interface in a separate window to see both the information panel and the visualization panel of the interface. Fig. 7 and Fig. 8 show the full interface and the smaller interface on the wiki page, respectively.

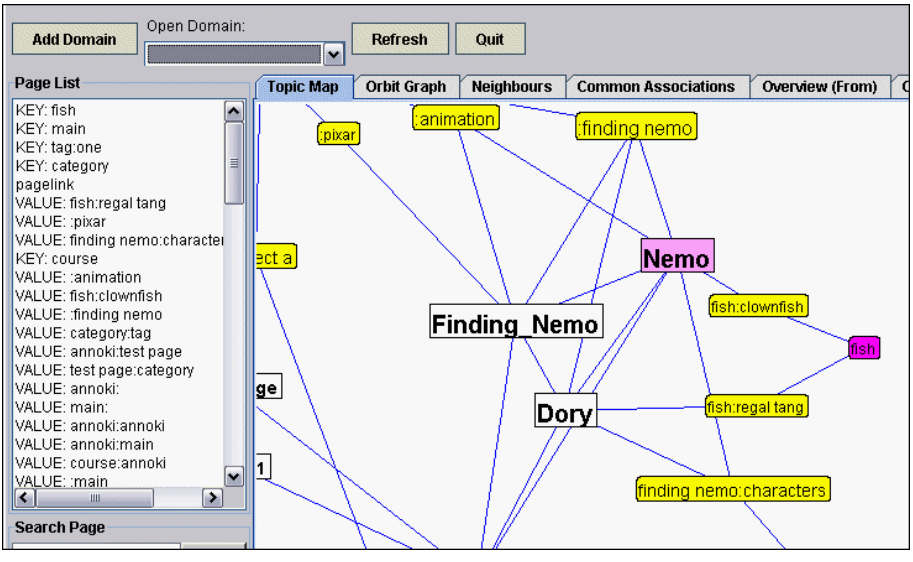


Fig. 7. The full Annoki Visualization interface

When the Visualization interface is launched, it first uses the Topic Map Generator URL to generate the XTM feed, which is used as the input for the visualization.

5 ENWiC Evaluation

To evaluate ENWiC, we carried out an experiment that tested its usability against a standard Wiki. We set up two conditions: the control condition, where the subject uses a standard Wiki to answer questions and the experimental condition, where the subject uses ENWiC to answer questions. Half of the subjects switched from the control to the experimental condition, and vice versa for the other half.

The subjects were computing science students, and so we chose design patterns as the subject domain for our tasks. We selected the Portland Pattern Repository's

Wiki [19] for this experiment since its content concentrates on software development, and we used ENWiC to crawl a mirror of the Wiki to create the topic map. We also used the Administrator component to put in hierarchy diagrams, double cell diagrams and comparison matrices to add some domain-dependent relations and keywords that the Crawler did not pick up in the pages. We did not modify the actual wiki pages; students under both conditions used the same content and information to get their answers.

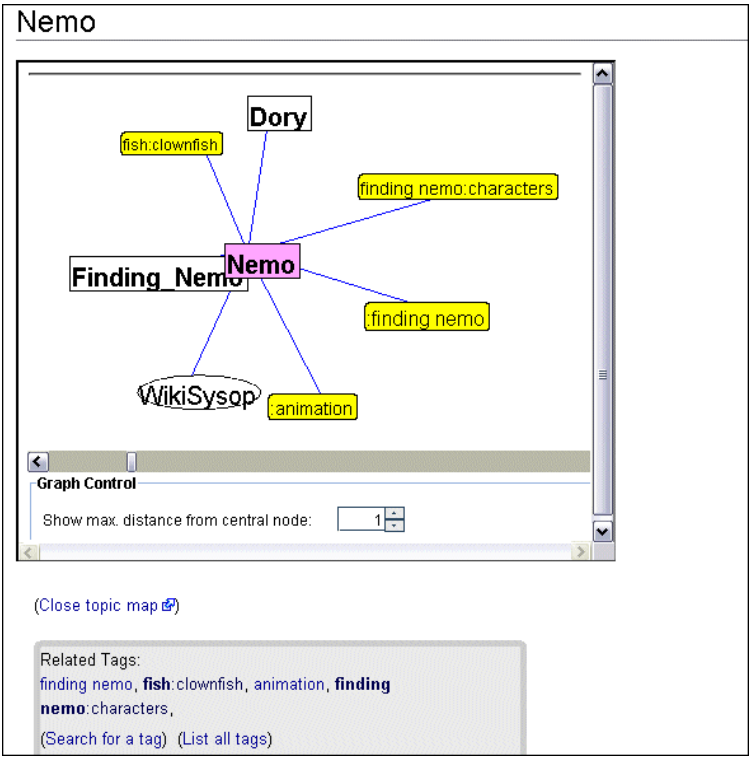


Fig. 8. The Annoki wiki interface together with the mini visualization interface

After giving them a short tutorial on Wiki and ENWiC, the subjects answered six questions: three using ENWiC and three using the Wiki. Each set questions contained one easy question, involving only one design pattern, and two questions that required subjects to combine information from different patterns. The subjects were given a maximum of 15 minutes to answer each question.

During the experiment, we observed the subjects and we recorded the time to completion, the number of unique nodes/pages visited, the number of mouse clicks on ENWiC nodes and the number of link traversals and back-button clicks on Wiki. After they completed their tasks, we asked our subjects to fill out a questionnaire about the usability of the two applications, and their overall impression and experience with the two applications.

Finally, we evaluated the correctness and the completeness of their answers.

Table 1. The ENWiC experiment tasks

easy	1. Give an example of how credit card service can use the Strategy pattern.
	2. Consider a chat room application. What is the subject and what are the observers?
difficult	3. What is the class of design pattern that Decorator belongs to? What is the class of design pattern that Memento belongs to?
	4. List two patterns that promote decoupling by using a central object for communication.
	5. Consider a pager service. Why is the façade pattern inadequate for this scenario? Which pattern would be more suitable?
	6. You would like to make a deposit to your personal accounts at the bank. The bank has a group of tellers; you are called to the teller that can handle deposit to personal accounts, who will process your request. What pattern does this resemble?

5.1 Results and Observations

We found that time-to-completion was not an adequate measurement since it depended more on the subject than the condition. Generally, completion times were shorter with ENWiC, but some subjects took more time than others regardless of the conditions. Also, subjects took less time to answer later questions since they remembered information from the first few questions and had a better idea of where to look. On average, completion time for a task in ENWiC was 3:17 minutes, while in Wiki it was 3:31 minutes. Since ENWiC's loading time per page is quite longer than Wiki, the completion times for ENWiC could potentially be even better.

For a better measure, we examined the number of unique pages/nodes visited and the number of clicks. As shown in Table 2, we noticed our subjects used significantly less number of clicks to complete the hard questions when using ENWiC. This was not significant with the easy questions where the answers were on a single page, but on the hard questions, subjects were able to move to desired pages just by using the graph, as opposed to having to click on the Back button many times while using a traditional browser.

Table 2. Number of mouse clicks and unique pages used by each subject (A–F) for each question (1–6)

	1	**2**	**3**	**4**	**5**	**6**
A	5,E:3,2	2,W:5,5	1,W:8,6	4,E:1,1	6,E:7,3	3,W:43,26
C	2,W:4,2	5,E:3,3	4,E:5,3	1,W:40,14	3,W:33,17	6,E:3,2
E	2,W:2,2	5,E:3,3	4,E:4,4	~~1,W:24,7~~	~~3,W:9,4~~	6,E:2,2
B	5,W:4,3	2,E:1,1	1,E:8,7	~~4,W:19,16~~	~~6,W:29,18~~	3,E:6,6
D	2,E:3,3	5,W:2,2	4,W:10,6	1,E:3,3	3,E:3,3	6,W:9,7
F	2,E:4,3	5,W:2,2	**4,W:2,2**	1,E:13,8	3,E:4,4	6,W:2,2

Each cell (formatted as O,W/E:C,P) reports the order in which the question was asked (O), the tool in which it was answered (E for ENWiC and W for Wiki), the number of clicks involved (C) and the number of unique nodes/pages (P) visited. The first three subjects in the table used Wiki first and then switched to ENWiC, while the other three subjects used ENWiC first before using Wiki. The crossed-out values indicated that the subject was timed-out before he could finish the task.

The numbers for the easy questions (1 & 2) were similar for both tools, but for the harder questions, it can be seen that the number of pages/nodes the subjects used in Wiki is more than ENWiC except in one case (outlined in bold), where that particular subject knew which pages to look. In terms of the number of clicks, the margin between ENWiC and Wiki is even higher, indicating that the subjects went through significantly less irrelevant information and pages when using ENWiC.

According to our observations, reading times were also reduced in some cases, especially when the subjects had to seek out certain relationships, as it was faster to look at the links on the graph than to read the text to find the information. Also, subjects using Wiki may overlook links to potentially better pages; for example, when answering question (1) from Table 1, the Wiki page that described Strategy pattern contains a link to another page that has an example of Strategy pattern. On ENWiC, that link has a section node "Example" associated with it, and thus ENWiC users easily recognized that link as a useful; it took Wiki users longer to realize that, and in some case they did not visit the link at all.

ENWiC was able to give the area of where solutions may be located. For example, when subjects answered questions (4) or (5) from Table 1, when they search and locate the Façade pattern, they identified the Mediator pattern as the most likely solution for the second part of the question since it appeared in the graph near the Façade pattern node; this was not as obvious for the Wiki users. When a subject misses such relationship, they have to go through more patterns to finally identify the Mediator pattern. Thus, ENWiC helps reduce the number of unique pages that a user has to read since it can limit the area of pages that user has to look through.

We carried out a statistical hypothesis testing based on our data. For each question, we examined whether the number of clicks and the number of pages visited for each task was smaller when using ENWiC than when using the standard Wiki interface. Thus, we state our hypotheses as:

$$H_0 : \mu_{Wiki} = \mu_{ENWiC} \text{ (Null Hypothesis)} \quad (1)$$

$$H_1 : \mu_{Wiki} > \mu_{ENWiC} \text{ (Alternate Hypothesis)} \quad (2)$$

$$\text{Selected significance level: } t_{0.05,4} = 2.132 \quad (3)$$

We used T-statistics for our hypothesis testing. The results are summarized in Table 3.

Table 3 shows that there was a significant decrease in the number of clicks and number of visited pages when using ENWiC for questions 4 and 5. These were difficult questions, in that to answer them the subjects had to combine information from more than one wiki pages. Question 3, although it was also difficult, did not have a significant difference because some subjects were able to utilize the Wiki Category page to answer the question. Question 6, again difficult, did not have a significant difference because some subjects remembered some information while looking for

solutions to previous questions. They were able to identify which pattern is the most likely candidate for the solution, thus reducing the number of pages and clicks they used while working on this question.

Table 3. ENWiC hypothesis testing result for each task. A value marked with * denotes it is significant enough to reject the null hypothesis.

Question	T_o (# of clicks)	T_o (# of pages)
1	0	-0.707
2	0.555	0.5547
3	0.372	0
4	*2.997	*2.428
5	*2.526	*2.138
6	1.127	1.121

We also carried out a similar hypothesis testing for the number of clicks and pages needed to complete all tasks. We believe that the number of pages and the number of clicks needed to complete all tasks in Wiki is more than in ENWiC. We state our hypotheses as:

$$H_0 : \mu_{Wiki} = \mu_{ENWiC} \text{ (Null Hypothesis)} \quad (4)$$

$$H_1 : \mu_{Wiki} > \mu_{ENWiC} \text{ (Alternate Hypothesis)} \quad (5)$$

$$\text{Selected significance level: } t_{0.05,34} = 2.032 \quad (6)$$

The results for the above hypothesis testing are summarized in Table 4.

Table 4. Hypothesis testing result for all tasks. A value marked with * denotes it is significant enough to reject the null hypothesis.

Question	T_o (# of clicks)	T_o (# of pages)
1-6	*2.611	1.790

The results in Table 4 validate our observations that subjects generally use fewer clicks when using ENWiC instead of Wiki. The shorter completion times and the lower number of clicks in ENWiC agree with the cognitive theory, which suggested that textual presentations cause more memory stress than visual presentations. Since it was easier for subjects to remember their location with ENWiC, they could complete tasks with shorter times and paths.

Correctness and completeness of the tasks were not significantly different between the two tools, which were expected because ENWiC gets its data from the Wiki, so once subjects locate the proper information, they usually arrive at the same conclusion. ENWiC just helps subjects to identify the proper data faster.

Since ENWiC does differ from conventional browsers, we found that subjects needed some time to learn and adjust to the interface. Some subjects learned quickly and were able to utilize the tool very well, while some had difficulties adjusting and learning how to read the graph and utilize the diagram filters. We found that subjects who like to use index and category pages from the Wiki were better at interpreting the

graph of ENWiC, while subjects who use the search function or prefer to read text may not adjust as well to the graph. Since the search functions in the Wiki and in ENWiC were not as sophisticated as search engines like http://www.google.com, the answers cannot be directly obtained using the Wiki or ENWiC search. Some subjects did appreciate that the ENWiC search function can narrow down the area of where the solution may be, but the few subjects who did not utilize the graph did not find ENWiC a lot more useful than Wiki.

Based on the questionnaires, almost all subjects found ENWiC to be an interesting tool to use. The main advantages they outlined were: relationships between topics can be seen clearly, which allows for easier navigation; they did not feel as lost when searching for answers; and they can select and traverse the relevant path faster.

The one disadvantage of ENWiC that all subjects agreed upon is its speed. Since the ENWiC viewer has to render a graph while loading a page, the display of it takes more time as compared to the standard Wiki interface. Subjects who are used to moving and reading pages quickly found the speed of ENWiC an annoyance that deteriorates satisfaction.

Multiple intelligence [8] suggested that there are eight different types of intelligence, which implies that there may be eight different styles of learning depending on the individual. Two of the eight types of intelligence are: linguistic (words) and spatial (visual). From our experiment, it is evident that ENWiC is useful for subjects who could interpret graphs and visuals better, as ENWiC is targeted for spatial learning, while subjects who learn with words (linguistic intelligence) may or may not find ENWiC better than a standard Wiki. Subjects who could adapt to both types of learning style liked ENWiC because it was it was different than conventional text navigation, which made it more interesting and enjoyable to use.

6 Conclusion

E-Learning is fast adopted as a means to support teaching and learning by educational institutions. Wikis are on-line knowledge bases, built collaboratively by communities of people studying or having an interest in a particular domain. As such they are frequently used in support of teaching and learning.

The ViskiMap visualization tools offer an alternative user interface to the standard text interface of Wikis. Its graphical representation of a Wiki helps users to see an overview ‘map’ of the Wiki, which enables efficient navigation. It is highly beneficial for users who are visual learners and like to see the relationships between topics, and its stimulating interface can capture users’ interest to encourage them to learn.

Our usability study indicates that ENWiC supports shorter times for accomplishing tasks based on wiki information. Through its visualization of the domain, ENWiC makes visible the wiki navigation structure, which is beneficial to self-learning. Its usage is not be limited to Wikis only, as its design can be easily applied large scale websites, such as blogs or online journals, which will help expand content that can be effectively used for e-Learning.

We plan to further investigate the use of wiki visualizations in e-learning with a longer experiment with Annoki.

References

1. Amin, A.B.M.: Using Graphic Organisers to promote active e-learning. In: Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005, pp. 4010–4015. AACE, Norfolk, VA (2005)
2. Aumueller, D., SHAWN: Structure Helps a Wiki Navigate. In: Proceedings of the BTW-Workshop WebDB Meets IR (2005)
3. Bloom, B.S.: Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. Addison-Wesley, Reading (1956)
4. Campanini, S.E., Castagna, P., Tazzoli, R.: Platypus Wiki: a Semantic Wiki Wiki Web. In: SWAP, Proc. Semantic Web Applications and Perspectives- 1st Italian Semantic Web Workshop (2004)
5. Canas, A.J., Carff, R., Hill, G., Carvalho, M., Arguedas, M., Eskridge, T.C, Lott, J., Carvajal, R.: Concept Maps: Integrating Knowledge and Information Visualization. In: Knowledge and Information Visualization 2005, pp. 205–219 (2005)
6. Dicheva, D., Dichev, C., Wang, D.: Visualizing Topic Maps for e-Learning. In: ICALT 2005. Fifth IEEE International Conference on Advanced Learning Technologies, pp. 950–951. IEEE Computer Society Press, Los Alamitos (2005)
7. Donelan, C.I.: From Spider Maps to Double-Cell Diagrams: Graphic Organizers Support Student Learning. In: ENC Online, http://www.enc.org/features/focus/archive/graphic/document.shtm?input=FOC-003559-index
8. Gardner, H.: Frames of Mind: The Theory of Multiple Intelligence, Basic (1983)
9. Lamb, A.: Learning Resources: Graphic Organizers. In: The Teacher Tap, http://eduscapes.com/tap/topic73.htm
10. Lee, Y.J.: Facilitating Web Search with Visualization and Data Mining Techniques. In: Knowledge and Information Visualization 2005, pp. 326–342 (2005)
11. O'Neil, M.: Automated Use of a Wiki for Collaborative Lecture Notes. In: Proc. of the 36th SIGCSE technical symposium on Computer science education, St. Louis, Missouri, USA, pp. 267–271 (2005)
12. Robinson, D.H., Skinner, C.H.: Why Graphic Organizers Facilitate Search Processes: Fewer Words or Computationally Efficient Indexing? Contemporary Educational Psychology 21, 166–180 (1996)
13. Smith, M.A., Fiore, A.T.: Visualization components for persistent conversations. In: Proc. of CHI 2001 (2001)
14. Stroulia, E., Jari, K.: EduNuggets: an intelligent environment for managing and delivering multimedia education content. In: International Conference on Intelligent User Interfaces 2003 (2003)
15. Wang, C., Turner, D.: Extending the Wiki Paradigm for Use in the Classroom. In: Proc. Of the International Conference on Information Technology: Coding and Computing, Las Vegas, Nevada, pp. 255–261 (2004)
16. Javapedia, http://wiki.java.net/bin/view/Javapedia/WebHome
17. MediaWiki, http://www.mediawiki.org/
18. Open Clip Art Library Wiki, http://www.openclipart.org/wiki/?title=Main_Page
19. Portland Pattern Repository's Wiki, http://c2.com/cgi/wiki?WelcomeVisitors
20. Portland Pattern Repository's Wiki Pattern Roadmap, http://c2.com/cgi/wiki?WikiPagesAboutWhatArePatterns
21. Semantic MediaWiki, http://wiki.ontoworld.org
22. Topic Maps, http://www.topicmaps.org/xtm/1.0
23. TouchGraph LLC, http://www.touchgraph.com
24. Wikipedia, http://www.wikipedia.com

A Simulation Environment to Assess Driving Performances while Interacting with On-board Telematics Systems

Gennaro Costagliola, Sergio Di Martino, and Filomena Ferrucci

Dipartimento di Matematica e Informatica, Università degli Studi di Salerno
via Ponte don Melillo, Fisciano (SA), Italy
{gcostagliola,sdimartino,fferrucci}@unisa.it

Abstract. The evaluation of user interfaces for vehicular telematics systems is a challenging task, since it is necessary to understand the effects of interaction on driving performances. To this aim, in 2005 we developed and presented a framework specifically conceived for the indoor evaluation of these systems. In this paper we present some significant improvements of that proposal. In particular, we describe a graphical analysis tool able to provide a clear and deep insight about driver behaviors using the high amount of data generated by the simulator. Moreover, we report on the evaluation analysis that has been performed to assess the effectiveness of the framework for measuring driving performances.

1 Introduction

Researches on User Needs report that people want to be connected any time and any place, even in their cars [13]. To answer this requirement, information and communication technologies have been fitted into automobiles, giving rise to *Vehicular Telematics Systems* (VTSs), which represent the in-vehicle convergence of mobile communications and information processing, allowing drivers and passengers to stay in contact with the world outside their car. VTSs allow drivers to exploit a plethora of features, such as multiple audio sources (MP3, DAB, and DVD), web browsing, e-mails, phone calls, voice control, and so on. Moreover, most advanced VTSs (e.g.: BMW *iDrive*, Fiat *Connect+* or GM *onStar*) are starting to provide *services*, i.e. advanced functionalities involving interaction with a support centre. Typical examples are remote vehicle diagnosis, dynamic route calculation (taking into account road, vehicle, traffic and weather conditions), tele-aid, hotel reservations, etc…

However, the diffusion of VTSs is also causing concerns about road safety, since these systems can heavily increase driver's mental distraction [3], [22], which is widely recognized as the most prevalent cause of crash [23]. Thus, since road safety is paramount, it is a short term priority to limit driver distraction inducted by VTSs, by enhancing the usability of these systems.

Human-Computer Interaction in the automotive domain can be considered a new and open research area [12]. Specific tools and approaches are required, mostly to assess the visual/cognitive workload inducted by these systems and to understand the effects of VTS interaction on driver-vehicle performances.

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 439–451, 2008.

Currently, several universities, companies and research centers, have equipped laboratories with sophisticated driving simulators, able to simulate a high variety of physical phenomenon, ranging from the kinematics effects inducted by different suspension geometries, to very complex traffic scenarios (for instance, the NADS [14]). Some of them are used also for indoor evaluations of VTSs, but these laboratories usually cost hundreds of thousands of dollars and are very demanding to set-up [7], being prohibitive for small institutions specifically focused on HCI research.

Thus, there is a strong needing for simpler VTS evaluation systems, able in the meantime to assure high-quality reports. To address this issue, in (Costagliola, 2005) we proposed a framework specifically conceived for the indoor evaluation of VTSs User Interfaces (UIs). The goal of this framework was to support researchers in an easy collection of valuable data on driver's behaviors (and thus on mental workload), being in the meantime cost-effective, by requiring standard hardware and simple set-ups.

In this paper we describe some improvements of the framework and report on its evaluation. One of the most distinguishing features offered by the proposed framework was the possibility to assess the navigation assistance provided by VTS, by automatically generating virtual test tracks starting from a VTS cartographical database. At the best of our knowledge, currently no other simulation facilities offer this specific but important feature, compelling to evaluate on the field the VTS navigation assistance module. In the following we will describe how we have improved the realism of generated tracks starting from a VTS cartography. Indeed, the current version of the system is able to reproduce much more realistic scenarios, taking into account road types, contextual information, etc… As a result, the realism and the sense of immersion in the virtual scenario experienced by the subjects, that represents a crucial feature for a simulation environment, has been improved.

Secondly, to handle the huge amount of experimental data collected per session, we developed an apposite application, named *Telemetry Analyzer Tool* (TAT). The aim of this tool is to support the data analyst in understanding the relationships occurring between VTS interactions and driving performances, by providing a clear, graphical representation of the subject behaviours, such as input on controls, followed trajectory, etc…, related to the timeline of the test. This is a highly demanded feature [1], leading to the generation of high-quality, repeatable analysis and reports.

Moreover, we have carried out an evaluation analysis meant to verify the effectiveness of the proposed approach for measuring on-road driving performances. To do that, we have employed a set of sixteen subjects. The conducted tests report that after about 20-25 minutes of training, subjects behave on the simulator in a way similar they drove a real car, thus supporting the validity of the simulator.

The remainder of the paper is structured as follows. In section 2 we introduce the main aspects to consider when dealing with driver distraction, as well as the approaches used to evaluate it. In section 3 we briefly recall the main characteristics of the evaluation framework, then we will focus on how we integrated the driving simulator and the vehicular navigation module. In section 4 we describe the data analysis features we developed, while in section 5 we report on the assessment of the framework. Finally, a discussion on final remarks and future work will conclude the paper.

2 VTS User Interface Evaluation Issues

Usability evaluation of traditional desktop applications can be considered an established topic within HCI. It is based on a shared understanding of basic concepts and extensive guidelines (e.g. [15], [20]).

However, these established concepts, methodologies, and approaches in HCI are being challenged by the increasing diffusion of *ubiquitous* computing, i.e. applications executed on wearable, handheld, and mobile computing devices. This move beyond stationary use is requiring new approaches to evaluate mobile UIs. Indeed, the main difference is that mobile systems are typically used in highly dynamic contexts, where the user is normally busy in other primary tasks, such as walking, driving, etc… [11].

This holds especially for the evaluation of VTS UIs. Indeed, static evaluations of these interfaces, performed with a subject totally focused on the system, do not provide significant information about the effectiveness of the UI. Instead, it is necessary to set up a meaningful test-bed, where subjects are mainly focused on the primary driving task and concurrently interact with the VTS. Researchers can evaluate driver distraction through the analysis of some indirect indicators on vehicle dynamics.

To accomplish these tests, two approaches can be adopted:

1. The interaction with a VTS is analyzed while the user is driving a real car (eventually on a track closed to the traffic), or
2. The driving is simulated in a laboratory, through some real-time computer-generated virtual scenarios.

Each of the two approaches presents advantages and drawbacks. The former is more realistic, because the subject drives a “real” car. However, it requires the availability of a car equipped with specific instrumentation able both to capture information such as travel speed and lane position and to video record the road scene and driver eye glance (e.g. [21]), and possibly of a closed track. Moreover, usability evaluations on the field are not easy. Three fundamental difficulties are reported in the literature. Firstly, it can be very complicated to establish realistic cases capturing key situations in the dynamic context above described [15]. Secondly, it is far from trivial to apply established evaluation techniques, such as observation and think-aloud when an evaluation is conducted in a field setting [17]. Thirdly, field evaluations complicate data collection and limits researchers controls, since subjects are acting in an environment with a number of unknown variables potentially affecting the set-up [8]. In particular, the last issue heavily applies to naturalistic tests in the vehicular domain, since data are usually collected by some video cameras, and many studies report how time-consuming and labor intensive is gathering data from these videos [1], providing in the meantime low-quality reports.

On the contrary, driving a virtual car, simulated by computer graphics in a laboratory, significantly reduces the above difficulties, since tests are accomplished in a safe and controlled environment, where the risk of personal injury and property damage is eliminated. Moreover, it is more comfortable for researchers, which can get a higher amount of high quality data and carry out more controlled and repeatable tests, by presenting to different subjects the same scenarios. The main difficulty is to provide an adequate degree of realism in the virtual scenario, since car dynamics and activities

in the subject's physical surroundings can be difficult to recreate realistically [18]. Thus, in order to ensure significant results of the tests, it is necessary to develop simulation environments which exhibit a high level of realism.

3 The Proposed Test-Bed

The test-bed we proposed in [5] was aimed at presenting a realistic simulated driving environment to conduct indoor evaluations of VTS UIs, by providing subjects with credible feedbacks for their actions (e.g.: steering wheel shake when leaving the lane and going off-road), and sense of presence in the virtual environment, being in the meantime cost-effective both in terms of hardware and personnel resources. The test-bed is intended as an integrated framework, composed of three main facilities: a driving simulator running on a graphical workstation, a VTS, and some tools to analyze subject's behaviours, stored in a telemetry database.

In order to provide driving features, we customized an open-source car simulation engine, the *Racer* system [24]. We choose this engine thanks to its many advantages: it provides satisfactory dynamics of the virtual vehicle by using 6 Degree-Of-Freedom models and motion formulae from Society of Automotive Engineers, it is very flexible, since almost all simulation parameters are customizable through text files, there is enough documentation, it supports force-feedback devices, it provides high-quality OpenGL rendering, the tracks and the scenes are quite easy to create both by scratch or through many free user-friendly editors, and last but not least, it is free for non-commercial use.

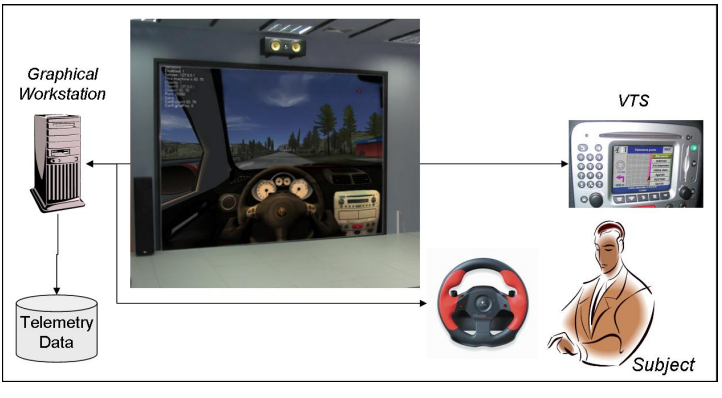


Fig. 1. The test-bed Architecture

Special care has been devoted to enhance the sense of presence of the subject in the virtual environment. Subjects sit on a car seat, interacting through a force-feedbacks steering wheel, able to return realistic haptic feedbacks to driving input, and two foot pedals. The simulated scenario is projected onto a wide-screen covering a significant subject's angle of view (>135°), while a 5.1 surround system provides a realistic spatial audio, as suggested by [7].

To get data able to inform on driver distraction, the simulator was customized to offer some further features, such as the possibility to generate asynchronous external events to test driver's response times. For instance, to add meaningfulness to the simulation, other simulated cars were added on the track with their own (repeatable) behaviors. To recreate these situations, we exploited the *Racer* features related to the Artificial Intelligence (AI), allowing for a basilar simulation of traffic conditions, programming different vehicles to follow specific routes and behaviors on the track.

The simulation engine has been enriched and complemented by some tools we have developed specifically to address vehicular issues. The former tool, allowing for an integration of the VTS navigator module in the simulated environment, is described in the following.

3.1 Integration of the Navigator in the Simulated Environment

The first distinguishing feature offered by our framework is the integration between the simulation engine and the VTS navigator module. Currently, at best of our knowledge, evaluations of the navigation assistance subsystems can be performed exclusively on the field, since virtual tracks employed in simulations usually are not a counterpart of a real geographic area. Even if dealing with digital reconstructions of existing roads, driving simulators do not provide features for converting in real-time the spatial coordinates of the virtual car into real GPS ones. Hence the navigator has not knowledge about the virtual car position, resulting of no use in indoor evaluations.

To address this issue, we developed two subsystems: one responsible to generate a Racer track stating from a real cartography, and one able to translate the coordinates of the virtual car driven by the subjects, into a stream of bytes emulating a real GPS serial sensor sent to the VTS. As a result, the road virtually driven by the user on the simulator is shared as a map on the VTS. This permits to exploit, indoor, many standard navigation features, such as Map Display and Route Guidance. This integration is a powerful instrument, enabling to perform many significant tests. For instance, it allows researchers to evaluate different modalities (vocal, iconic, etc…) for providing routing assistance to the user, or the most appropriate vocabulary to support the way-finding, as well as to assess the cognitive work inducted by these different approaches.

In the following we will describe how we have developed this feature.

Sharing the Cartography

The Racer engine adopts a proprietary graphical format to represent tracks, named DOF1 and based on the SGI IFF file format. DOF1 exploits OpenGL XYZ coordinate system and contains all the information about the scene graph of the model. In particular, it holds data about the geometry objects composing the track, i.e. information about the vertices and the normals, together with other data, such as the texture used to render the surfaces.

It is worth pointing out that currently there exist two standard global cartographical databases adopted in the automotive domain: the Navteq and the TeleAtlas one. We focused on the former, thanks to the availability of a powerful SDK, useful to manage navigation system applications and to interact with the Navteq open format SDAL [19] used for the map database.

In order to obtain a shared cartography, we developed a tool able to generate an appropriate DOF1 track file starting from an arbitrary sized rectangular area of a SDAL map. In particular, this translator generates the geometry primitives starting from the parcels that are the basic units of I/O used in the SDAL format. Fig. 2 (left) shows a rectangular area of a SDAL map, while Fig. 2 (right) illustrates the corresponding generated DOF1 track.

Another challenging issue was about road width, since SDAL does not contain such information. Instead, for each segment representing a section of a road, there is an associated attribute, the *Rank*, indicating its rating, based on characteristics such as speed limit, road type (e.g., interstate, highway) and access limitations. For instance, rank 0 represents the lowest one (local streets), while rank 4 represents the highest level (national arterial road network). Basing on this information we defined 10 kinds of road templates (two for each rank, to consider one or two-way roads), each one with its own geometries and textures to represent different number of lanes, presence of guard-rails, etc…

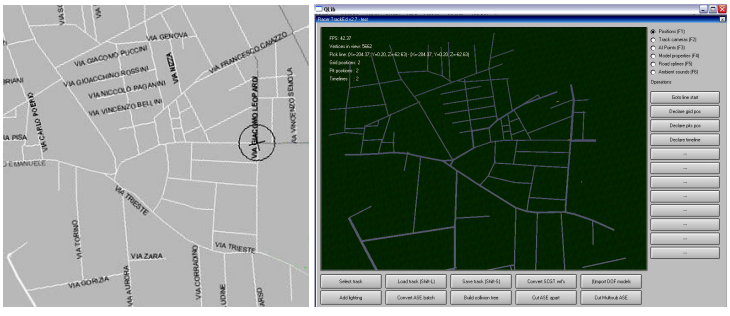


Fig. 2. A real SDAL map (left), and the corresponding generated DOF1 one (right)

Updating the Localization

To address the second issue concerning with updating in real-time the position of the car on the map we have let the simulator to export information about the car movements and the navigator to accept this information as if it comes from GPS sensor. In particular, it was required to get information about coordinates, speed, and heading of the car. *Racer* engine is able to output this information, for various purposes (multiplayer, logs, etc…). Thus, we implemented a daemon able to catch this information and, after some elaboration, to send the necessary data on a serial port, emulating a real GPS sensor. In this way, every VTS (or even PDA) able to be connected to a GPS serial sensor can be linked to the simulation environment, providing real-time route guidance information.

3.2 Data Analysis

In order to assess a VTS UI, it is important to quantify the "safety degree" of the considered VTS. Nevertheless, safety cannot be directly measured (probably except in retrospect) [22]. In the literature, several indirect measures of safety have been proposed

based on the evaluation of the driver distraction inducted by the system (e.g. [4]). Let us recall that, when dealing with vehicles, two main kinds of distraction should be considered, namely the visual one and the cognitive one. Each of them leads to different problems: degraded vehicle control (resulting in problems in lane-keeping, speed maintenance, etc…), and degraded object/event detection (*looked-but-did-not-see*) [2], which is a more insidious to evaluate, since vehicle control remains largely unaffected but detection and reactions of unexpected object and event is degraded [22]. Several indicators have been proposed to measure driver distraction during a test session. Among these, the most important are *speed maintaining*, *input smoothness*, *lane-keeping* (see Fig. 3), *car following performance*, and *driver reaction times to asynchronous events* [22].

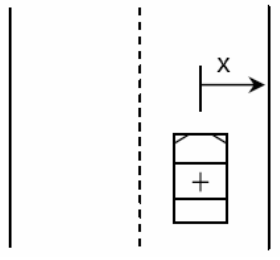


Fig. 3. Distance to be measured for Lane-Keeping

It is worth noting that the result of a VTS usability assessment is a set of *telemetry* data, i.e. a lot of numerical information, such as user input and vehicle dynamics, describing how the car and the driver behaved during the experiment. It is important to properly analyze this valuable amount of data in order to get information on subjects' distraction and highlight potential degraded vehicle controls or degraded object/event detection. Moreover, these telemetry data can allow researchers to infer how subjects' visual/cognitive workload is influenced when external factors are changed. For instance, it is possible to understand the consequences of exploiting different sensorial channels or different layouts for VTS graphical user interfaces, by comparing the gathered test datasets on the same subjects. In order to properly analyze these data, it is necessary provide analysts with suitable tools supporting them during information interpretation. To this aim, the proposed framework has been enhanced with a specific data analysis environment, detailed in the following.

4 The Telemetry Analyzer Tool

The Racer provides a built-in logging feature, which was customized to gather all the needed telemetry data. In particular, during a test running, each 250 milliseconds we sample the following parameters:

- Time
- User Input
  - Steering input
  - Force Feedback output

  - Throttle input
  - Brake input
- Vehicle Dynamics
  - yaw, pitch and roll
  - vx, vy and vz
  - ax, ay and az
  - x, y and z
  - Overall distance
- VTS I/O (if possible)
  - Key pressed on the VTS prototype
  - Asynchronous events generated by VTS

All these data are stored in a log ASCII file. Separate files are used to store other significant information, such as details on asynchronous events generated both by the simulator, (i.e.: AI controlled cars actions), and/or by the tester (i.e.: actions required to be achieved by the subject).The amount of data collected per session in this way can be really impressive. For instance, a 20 minute registration (a typical duration for a VTS test session) generates a data matrix with about 4800 rows and tens of thousand of cells. As a result, it is almost impossible to effectively analyze this huge but fundamental amount of data without a suited supporting tool. Neither standard analytics software products, such as *R*, *Statistica* or *Excel* can successfully address this issue, since it is very difficult for a researcher to understand the most significant distraction indicator, the lane-keeping (shown in Fig. 3), without a graphical representation of the subject's followed trajectory overlapped to the testing track.

Thus, there is a strong needing for a supporting tool, able to graphically visualize the fundamental parameters describing subjects' behaviours. To answer this issue, we developed an integrated application, named *Telemetry Analyzer Tool*, whose main objective was to visually render the driver-vehicle performances during the whole test session. In particular, the tool graphically presents both the temporal/spatial relationships among data (that usually are difficult to understand in a textual way), both the main subjects' actions together with other numerical information, such as speed, time, and overall distance.

In the following the main features of the tool and its user interface are detailed.

4.1 Main Features of the Developed Tool

The tool offers three main features. Firstly, it visually present all subject's input, time by time. To control the timeline of the logged data, the tool provides some widgets, recalling an interface of a Video Recorder, allowing a researcher to gain a deep insight on subject's behaviours, through the whole test session.

Secondly, it provides a visual representation of the trajectory followed by the subject, overlapping the map of the track. To evidence driving errors, whenever the trajectory followed by the vehicle's centre of gravity comes within less than 50 cm of the left/right edge of the lane, it is represented in red colour (or in a different shape) for the specific frame.

Thirdly, it provides some comprehensive report on the test session, such as the percentage of time spent too near or too far from the central line, etc…

The resulting user interface is shown in Fig. 4.

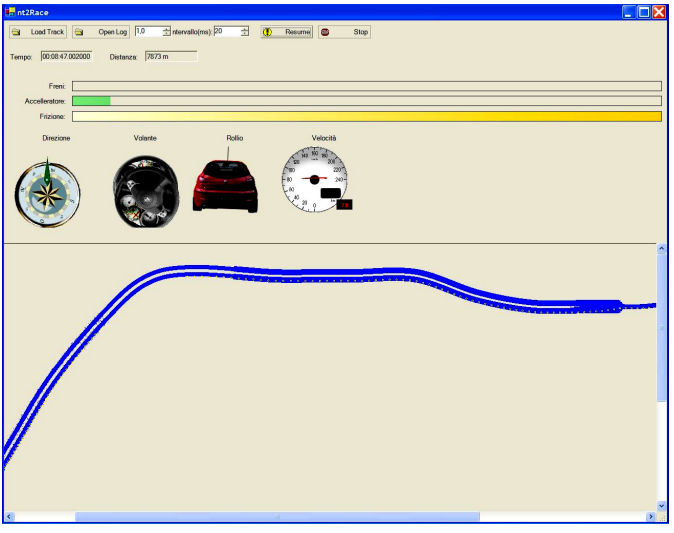


Fig. 4. The Telemetry Analyzer Tool GUI

As depicted, it is possible to notice that the UI is divided in three horizontal zones:

- The uppermost contains the widgets suited to load track and log files and to manage the timeline through some controls recalling a VCR. Through these controls it is possible to select a specific frame t_i, a time-interval t_i-t_j, or to playback the entire test session t_0-t_n. It also possible to change the time scale, to accelerate or slow down the playback, as well as to move frame-by-frame.
- The central section is aimed at dynamically present visual information about user behaviours, for the current frame t_i. Once selected an instant of time t_i, or for each frame during the playback, it renders the status of all user input at t_i. In particular, driver input on brake and throttle are rendered through some histograms, while the steering wheel angle is presented both visually and numerically. This section provides also information about main parameters on vehicle dynamics, such as heading, speed, and roll, useful to evaluate overall input smoothness.
- Finally, a lower section shows a top view of the track, with the trajectory held by the subject in the current frame/interval. This section also encompasses widgets suited to control zoom, pan, and rotation of the track.

About numerical reports, the tool is able to invoke the Microsoft Excel DCOM control to generate graphs of the collected data, on a number of different parameters. Some examples of generated scatter plots are visible in Figures 4 and 5.

5 Assessing the Proposed Test-Bed

As stated in section 2, the potential problem with an indoor simulator is the lack of realism. To this aim, we conducted a preliminary study to validate the effectiveness of

the simulator in measuring real driving performances, by involving a group of sixteen external subjects. To gain insight on the realism of the framework, we applied methodologies proposed in other simulators' assessments (e.g. [10]). In the following we provide the details on the assessment.

5.1 The Subjects

Sixteen volunteers (11 males, 5 females) were involved in the study. Age of the subjects varied between 24 and 42 years (mean 31.6). All participants have a valid driving licence, a good corrected visual acuity, and years of driving experience ranged from three to 21 (mean of 10.4). Annual mileage was between 2,400 and 25,000 (mean of 8,250). We asked them also details on typical driven roads, mean number of hours spent in driving per week, and if they usually play at driving computer games. Four subjects stated that they spent over 1 hour each week playing computer games that involved driving (racing). Two subjects reported no previous experiences with driving games. The other participants reported to play computer games only occasionally.

5.2 Methodology

The experiment consisted of three driving sessions, gradually increasing in difficulty. Whenever a sufficient level of proficiency was rated, subject move the next step. After completing the driving tasks, subjects received a questionnaire, to express their impressions on the simulator.

During the first session of the experiment, we tested elementary driving skills, such as steering, acceleration and braking control. Subjects were instructed to drive on a simple and flat track, composed of many straights, and few, easy turns, maintaining a stable speed of 60 km/h, if possible. Just before a curve, they were asked to reduce speed, to enable a smoother turning.

The second session of the experiment was based on a more complex track, representing a hill road, with turns, tunnels, climbs and descents hills, harder turns, and other cars. Again, we asked subjects to maintain a steady speed of 80 Km/h, and to never pass 100 Km/h, a harder task due to the specific characteristics of the track.

In the third session, subjects were asked to perform some secondary tasks on a VTS, while concurrently continuing to perform the primary driving task. The track was easier than the previous one. Subjects had to respect the signals, to keep a mean speed of about 80 Km/h, if compatible with track properties and speed limits, and to never pass 100 Km/h.

Approximately each session took 20 minutes per subject. The selected test tracks were intended as closed circuits, and each subject was asked to drive for three laps, to better understand improvements in vehicle controls.

Obviously, during each session we logged all the parameters described in the previous Section.

5.3 Results

We observed that the simulator has a smooth learning curve. Subjects familiarized very quickly with the simulator, reaching adequate performance after 15–20 minutes of training. In particular, after some initial hesitations due to the different steering

response, subjects were able to profitably drive the vehicle, maintaining constant speed and being able to correctly keep the lane. Moreover, these behaviours were achieved through smooth input.

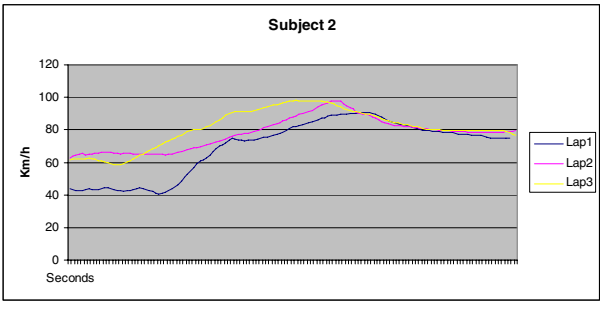


Fig. 5. Subject 2 driving behaviour

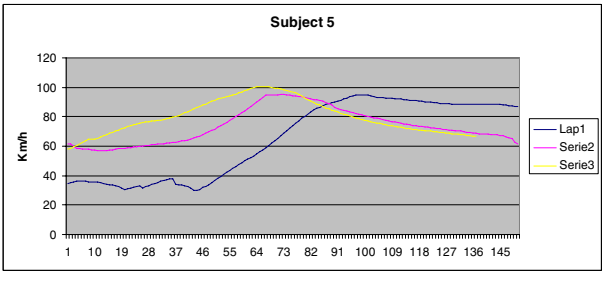


Fig. 6. Subject 5 driving behaviour

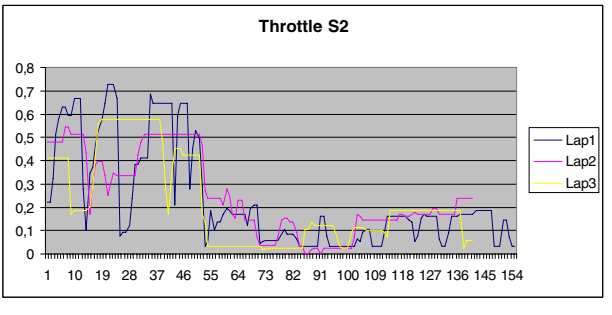


Fig. 7. Subject 2 throttle behaviour

A graphical representation of such improvements is given in Fig. 5 and Fig. 6. Such figures plot the speed hold by Subjects 2 and 5, respectively, in the same section of the track, on the three different laps during session 2. We selected for the graph a segment where the road descents, turn right and then climbs. Thus it is very demanding to maintain constant speed, and in particular to not overpass the 100 km/h.

Looking at graphs, it is possible to notice that in the first lap (the blue line), subjects were not able to smoothly maintain the speed. Indeed, the line presents steps, and, in case of S5, it varies from 30 to 95 Km/h. Lap 2 (the purple line) and 3 (yellow line) were smoother, indicating enhancements in driving performances.

Similarly, in Fig. 7 it is possible to appreciate improvements in the throttle control, through the various laps. Once completed the driving sessions, subjects filled in a questionnaire, to express their impressions on the simulator. About results, in mean, subjects felt the simulator enough realistic (6.57, in a ramp 0..10), even if more stressing and a little more difficult than real driving.

However, some concerns come from the steering wheel, which resulted too much direct in its action, being different from the real one. This is due to the economical input device, which is more game-oriented. Indeed, its excursion is limited to about 200° from full left deflection to full right deflection. In real car this value is comprised from 360° and 720°. We are currently searching for more realistic steering wheels. Telemetry data analysis and survey result are omitted for sake of brevity, but are available upon request.

6 Conclusions and Future Work

Safety on the roads is one of the main goals for everyone involved in the automotive field. The advent of VTSs can distract user from the main task of driving the car, with potentially fatal effects. Nevertheless, it has been estimated that these systems will become commonplace in the last few years. Thus, it is a short term priority to investigate solutions to enhance usability of VTSs and then limit driver distraction. Nevertheless, the evaluation of UIs for automotive systems is a challenging and expensive task, requiring specific methodologies and tools. To address this issue, we realized a framework specifically conceived for the indoor evaluation of VTSs usability. In this paper we reported on some improvements we developed, aimed at enhancing the effectiveness of that test-bed. In particular, we developed some instruments to offer the possibility to assess also the navigation assistance provided by VTS in the indoor facilities, thanks to the automatic generation of realistic simulator tracks starting from a VTS cartography.

Moreover, to support researchers in an easy collection of valuable information on driver's behaviors (and thus on his/her mental workload), we developed a specifically suited application, aimed at providing a graphical representation of the main driving parameters and subject behaviors. Finally, we validated the framework in measuring on-road driving performances, by employing a set of sixteen subjects, with positive results.

About future work, we are working to add further realism to the generated scenario, in order to recreate the surrounding environment, since SDAL contains further information about the kind of area (country, national park, urban, etc…). For instance, for urban zones, we are working to generate scenarios with buildings, semaphores, etc…

References

1. Barr, et al. Exploratory Analysis of Truck Driver Distraction Using Naturalistic Driving Data. 82nd Annual Meeting of the Transportation Research Board, Washington, DC, (January 12-16, 2003)
2. Brown, I.D.: Driver fatigue. Human Factors 36(2), 298–314 (1994)

3. Burns, C., Lansdown, C.: E-Distraction: The Challenges for Safe and Usable Internet Services in Vehicles (2001)
4. CAMP. Crash Avoidance Metrics Partnership (CAMP), Proposed Driver Workload Metrics and Methods Project (2000)
5. Costagliola, Di Martino, Ferrucci: A Framework for the Evaluation of Automotive Telematics Systems. In: ICEIS 2005. Proceedings of the 7th International Conference on Enterprise Information Systems
6. Green, P.: SAE J2364 - Navigation and route guidance function accessibility while driving. Society of Automotive Engineers, Warrendale, PA (1999)
7. Green, P., et al.: Audio-Visual System Design Recommendations from Experience with the UMTRI Driving Simulator, in DSC North America, Proceedings, Dearborn, Michigan (2003)
8. Johnson, P.: Usability and Mobility; Interactions on the move. In: Proceedings of the First Workshop on Human-Computer Interaction with Mobile Devices, Glasgow, Scotland, GIST Technical Report G98-1 (1998)
9. Juliussen, E.: The Future of Automotive Telematics. In: Business briefing: global automotive manufacturing & technology (2003)
10. Lee, H.: The Validity of Driving Simulator to Measure On-Road Driving Performance of Older Drivers. In: Proceeding of 24th Conference of Australian Institutes of Transport Research, Sydney (2002)
11. Lumsden, J., Brewster, S.A.: Paradigm Shift: Alternative Interaction Techniques for Use with Mobile & Wearable Devices. In: Proceedings of CASCON workshop (2003)
12. Marcus, A.: Vehicle User Interfaces: the next revolution. Interactions 1 (2004)
13. Microsoft. Microsoft Software and Solutions for the Automotive Industry (Last visited on 01 August 2007), http://www.microsoft.com/industry/manufacturing/automotive
14. NADS, National Advanced Driving Simulator (Last visited on August 01, 2007), http://www.nads-sc.uiowa.edu/
15. Nielsen, J.: Usability engineering. AP Professional, Boston, MA (1993)
16. Nielsen, J.: Testing in the Field. In: Proceedings of the third Asia Pacific Computer Human Interaction Conference, IEEE Computer Society, Los Alamitos (1998)
17. Pascoe, J., Ryan, N., Morse, D.: Using While Moving: HCI Issues in Fieldwork Environments. Transactions on Computer-Human Interaction 7(3), 417–437 (2000)
18. Pirhonen, A., Brewster, S.A., Holguin, C.: Gestural and audio metaphors as a means of control for mobile devices. In: Proceedings of CHI 2002, ACM, New York (2002)
19. SDAL File Format. Specification (1999) (Last visited on August 01, 2007) available at http://www.sdalformat.com
20. Shneiderman, B.: Designing the User Interfaces. Addison Wesley, Reading (1998)
21. Tijerina, L., Parmer, E., Goodman, M.: Driver workload assessment of route guidance system destination entry while driving: a test track study. In: Proceedings of the 5th ITS World Congress, Seoul, Korea (1998)
22. Tijerina, L.: Issues in the Evaluation of Driver Distraction Associated with In-Vehicle Information and Telecommunications Systems (2001)
23. Toms, M., Cummings-Hill, M., Curry, D., Cone, S.: Using Cluster Analysis for Deriving Menu Structures for Automotive Mobile Multimedia Applications. In: Human Factors in Automotive Design (2001)
24. Van Gaal: Racer Simulator Engine (Last visited on 01 August 01, 2007) available at: http://www.racer.nl

Handheld Tools for Personalized and Proactive Psychotherapy

Luís Carriço, Marco Sá, and Pedro Antunes

LaSIGE & Department of Informatics, Faculty of Sciences, University of Lisbon
Campo Grande, Edifício C6, Piso 3, 1749-016 Lisboa, Portugal
{lmc,marcosa,paa}@di.fc.ul.pt

Abstract. This paper presents a set of components that support psychotherapy processes on mobile and office settings. One provides patients the required access to psychotherapy artefacts, enabling an adequate and tailored aid and motivation for fulfilment of common therapy tasks. Another offers therapists the ability to define and refine the artefacts, in order to present, help and react to the patient according to his/her specific needs and therapy progress. Two other components allow the analysis and annotation of the aforementioned artefacts. All these components run on a PDA base. Evaluation results validated some of the design choices, and indicate future directions and improvements.

Keywords: Psychotherapy, Active Applications, Building UI, Usability, PDA.

1 Introduction

Cognitive Behavioural Therapy (CBT) is a common form of therapy used in the treatment of patients with anxiety and depression disorders. This constructivist therapy relies on the therapists' ability to understand the patient's problems and distorted cognitions, consequently guiding him to replace them with more adequate ones [12]. The therapist's work includes the main on-consultation activities (e.g. conversation and annotation), the definition of the adequate therapies and the analysis process. On the patient's side, the tasks involve the filling of the various forms, as a therapy, within or outside consultations (e.g. while working, at school, on the way home or whenever stressful situations arise). The resulting data also provides the therapist the information required to analyze and define further therapies and procedures, either on or between sessions.

Thus, the major challenges of the therapy process are: gathering patient's data; defining forms, questionnaires and suitable guidelines for each patient's specific needs; leading patients to perform tasks and register them; and analyze those registers and gathered data. Furthermore, these activities have to be coped with the constant need for patient/therapist collaboration and mobility.

Currently, used paper artefacts obstruct, in various ways, the therapist's work and, in some cases, a more rapid progression of the patients' conditions. Cross-referenced and structured annotations and expedite analysis, for example, are awkward or time consuming. Also, the passiveness of paper forms prevents the therapist from defining incentives and rewards that could allow patients to pursue their therapies in a more

Y. Manolopoulos et al. (Eds.): ICEIS 2006, LNBIP 3, pp. 452–464, 2008.

efficient way. These and other problems are recurrent throughout the entire process of diagnosis and therapy. Recently some work emerged on the introduction of technology on the process [1], [5], [8], [13]. However they tend to provide solutions for particular disorders, without therapist control and patient specificity.

In this paper, we present several prototypes which take advantage of the emerging mobile technologies. They were designed to be used during consultation settings or outside, in both patients' and therapists' daily activities. Together, independently or complementing each other, they support the multiple phases of psychotherapy, emphasizing the therapist's central role in the therapy adjustment and the cooperation with his patients. Besides providing both actors with common therapy tools, they aim to bring therapists the possibility to define specific artefacts, choosing their structure, content and even presentation modes. All the components are available in multiple platforms, although this article focuses on the PDA base. They are part of tool-set that was designed and developed within SCOPE project [4], involving computer engineers and psychologists.

The work described focuses one of SCOPE's contributions: the ability to create and refine psychotherapy forms that actively react to the patients' usage behaviour, to his/her specific needs, and that can be used throughout the day on any location.

The paper begins with an overview of the related work in this area followed by a description of the main requirements to support the psychotherapy process. The following sections describe our system and the different tools that compose it, as well as the details of the various components, with particular focus on those which intend to support the therapist's work. Afterwards, the evaluation that has been taking place and some of the results that have already been achieved are discussed. Finally, future work is delineated and some conclusions are drawn.

2 Related Work

The introduction of new technology to overcome some of the difficulties referred in this paper has recently gained momentum. As in many other areas of the health care domain, they often focus on data gathering or visualization, analysis and especially organizational tasks within the health care domain [7]. Specific software, directed to the psychiatric and psychological use, allows patients to follow particular methods of therapy and even diagnosis [15]. Excluding patient solutions that, relying on expedite approaches of diagnosis, have revealed strong human rejection [5], studies have demonstrated the effectiveness of the computer role in the process of anxiety and depression therapy [9], [10], [14], [20]. However, most of these systems provide either isolated therapist solutions or isolated patient solutions with no therapist control. Moreover, mostly rely on desktop approaches, which are incompatible [11] with most of the really used scenarios (e.g. work, school, office consultation).

A number of web-based self-help applications and websites are also emerging. For example, the use of an internet-based therapy for depression and anxiety caused by Tinnitius (the presence of sound in the absence of auditory stimulation), in which patients reproduce the face-to-face treatment [1]. Overall, in spite of the common advantages, such as remote assistance and costs lowering, these techniques have many disadvantages [19]. Patient disengagement is frequent, as well as patient misinterpretation of

the sites' objectives; sites require constant management and monitoring; and email and telephone assistance is often required.

More recently, handheld and overall mobile devices, such as PDAs or TabletPCs, are available, and new applications have appeared. However, they only cover partial steps of the therapy process and do not allow the customization of the patients' tasks or artefacts. The majority is rather simple and allows simple measurements of the severity of pathologies, indicates drug dosage or provides therapists with reference information about diseases or drugs [8].

On the patients' side, some self-control or relaxation procedures are available on hand-held devices [16]. Here, patients engaged on group therapies using palmtop computers. A palmtop was given to each patient, before the therapy sessions started, and had to be carried by him/her at all times. Alarms sounded several times during the day and questionnaires regarding their anxiety levels were popped. After the therapy sessions started, patients were allowed to access the entire range of features, using the relaxation, cognitive, etc. modules. Palmtops were intensively used during the initial stages of therapy, but their utilization diminished during the rest of therapy, which indicated that palmtops were mainly used to learn and understand the therapeutic procedures. Even so, patients improved their state in less than 6 months. But once again, these applications lack the possibility of adapting each step of the procedure or application to patients' specific needs.

3 Requirements

Constructivist psychotherapy methods, such as CBT, defend that humans are active participants in their own lives, agents acting and experiencing in the world. Based on this, individuals must be actively involved in their change and unfolding processes, and must be encouraged to be autonomous [12]. In therapy, the patients are therefore motivated to perform tasks that not only result in an active behaviour, but also permit them to be confronted with their accomplishments. Common tasks include:

- Planning activities, registering plans and the activities, registering thoughts and answering questionnaires.

These tasks are performed during the day, intermingled with other daily activities, triggered by them (e. g. a though requires a thought registration) or at specific times (e.g. at 9 am, plan the day). Accordingly they are performed wherever the patient is. Time, place and task characteristics, while based on the specific disorder, should naturally depend on the patient's life and severity state.

On the other hand, the problem often resides on leading the patient to do something or be confronted with something he/she does. Therefore, a central role of the therapy is also given to therapists, for their ability to get patients to act and gain conscience of their problems. To accomplish this endeavour the therapist must be able to:

- Gather patient data (e.g. what he/she says, or even what gesture was done) and annotate it.
- Diagnose the disorder characteristics.

- Select and define the therapy procedures and artefacts, motivating the patient and ensuring that that motivation is kept even when the therapist is away.
- Get the patient registered data and analyze it thoroughly, eventually comparing it with previous gathered data and annotations.

The therapy cycle completes a tour when therapist-patient conversations take place over the registered data and new events. These cycles (data gathering, diagnosis, prescription, task fulfilment and thorough analysis) actually involve both actors and usually prevail through a long period of time.

As for the patient, the therapist activities are often accomplished in different settings and times of the day. Motivation, data gathering and exchange are naturally mostly done in collaboration with the patient, within consultation sessions. Analysis, data comparison, diagnosis, therapy planning and prescription are usually done elsewhere, although they also occur often within a consultation setting. Therapy planning and prescription, for instance, could be adjusted considering the information just gathered in an on-going consultation session. Moreover, since fulfilment of patient tasks is frequently done or completed in collaboration with the therapist, during sessions, particularly in the initial states of therapy, that information can be of the utmost importance on the following therapy procedure.

3.1 Artefact Requirements

Although the procedures and therapists' instruments, in CBT, are well defined, the latter exist mainly in paper. This causes coding and organization problems, obstructs co-referencing and annotation, and most of all, wastes both therapist and patients' time. Furthermore, it prevents therapists from adapting the existing artefacts to the specific needs of each patient and on motivating him.

The used forms and questionnaires may be composed of hundred of numeric values, or, when of free writing type, unreadable documents, rather difficult to analyze. The notes taken during a single session or a set of sessions are also extremely valuable artefacts to analyze. However, it is difficult to co-relate these annotations, the questionnaires or even sessions they correspond to, especially having in mind that most of them are taken during consultations and without distracting the patient from the conversation or task in hands.

Paper artefacts obstruct patients' tasks as well. Questionnaires that need to be filled can be very extensive and with questions that may not be suitable for the patient in question. The free-text forms, generally preferred by the patients, because of their flexibility on writing and expressing their feelings, may cause distress as well, since, at times, the patient might not know how to express his sensations or problems in an organized and understandable way, increasing the anxiety levels. These problems suggested the following requirements:

- Minimize the effort and time spent taking, organizing and correlating annotations.
- Provide the therapist means to easily adapt the existing artefacts (e.g. forms, questionnaires and activity plans) to the specificity of patients or even create new ones.
- Offer the therapist ways to easily select and define rewards, filling facilitators (hints), help (on procedures and filling), and criteria that triggers them.

- Engender artefacts that can react to the patients' entered data and scheduled events, offering aids and rewards.
- Enable the analysis of patient completed questionnaires and forms, as well as therapist annotations.

3.2 Architectural Requirements

One of the major challenges of the psychotherapy activities is that they can be done anytime and anyplace, sometimes individually and other times in collaboration. Resources and interaction requirements depend on the situation. Moreover, throughout the process data is created or gathered that must be kept available and moved trough the different settings as needed. On consultation settings, for instance, actors exchange information, through conversations or defined and filled artefacts. All these clearly impose constraints, hardly handled by a paper based approach, which nevertheless must be considered on the definition of a technological solution. The following requirements are suggested:

- Enable the therapist to perform deep thorough analysis, on large amounts of data, but also permit to define prescription and simple analysis on the different settings of its daily activities.
- Provide means with no fixed or physical constraints that allow patients to accomplishing their tasks during their daily life.
- Offer ways that do not interfere with the face-to-face consultation processes and yet support different forms of collaborative work.
- Provide simple ways to transfer artefact definitions and information from its completion, between actors, and also to move information and articulate its access trough the different settings.

4 Scope

SCOPE (Supporting Cognition Outlines on Psychological Evaluation) is a project that aims at providing computational support to psychological therapy through all the steps of the process. It ranges from: (1) analysis, diagnosis and prescription tools, which the therapist uses without the patients' company; to (2) patients' homework registering tools, used without the therapist presence; through (3) a set of tools that they use in co-presence, and that include all of the above activities. Naturally, the characteristics of the face-to-face setting tools are different from those used individually, as they should not interfere with the collaborative process.

Figure 1 depicts the general settings of SCOPE: one presumes a full resource environment (called fixed) that usually comprehends a PC, possibly a patient record server, and peripherals (e.g. printer); another (referred to as mobile) copes with the mobility requirement with variable resource capabilities and may be supported by a laptop or a handheld device; and the last covers the co-presence situations (named face-to-face), where TabletPC and handheld solutions are envisioned. The first two presume, primarily, an individual stand where therapist and patient work isolated from each other - collaboration through a network is also possible but it is the subject of JoinTS [3], a continuation of SCOPE.

On these settings, some scenarios can be envisioned resulting in one or more applications. In general, each application may operate in stand alone mode, possibly integrating a classic paper based therapy procedure. On the other hand, applications can be articulated with each other, covering most of the therapy activities. Integration is accomplished through a shared database, exchanging artifacts and data through a synchronization process (represented by a strait-dashed line in the figure), or through direct communication (a curved arrowed line). Note that direct communication also occurs between therapist and patient in the face-to-face setting. Applications can also be configured (depicted by a fat arrowed symbol) to become more adequate to a specific usage scenario.

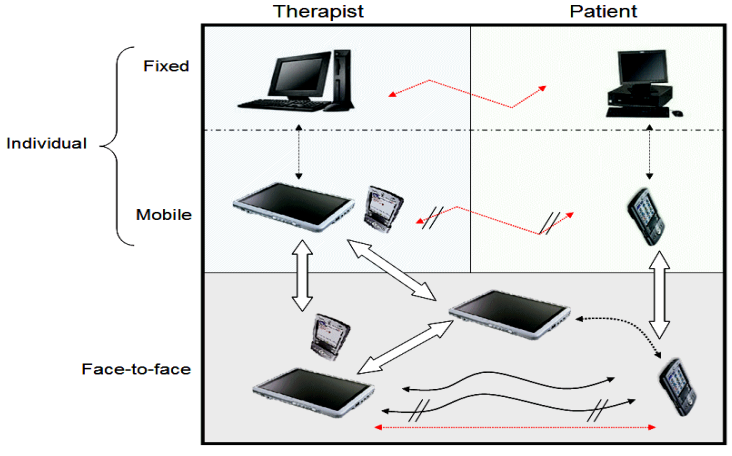


Fig. 1. SCOPE overall setting and scenarios

5 Therapist Applications

This section presents a more detailed description of the components of the SCOPE tool-set that pertain to the therapist. It focuses on the PDA platform, although variants of the tools are also available for the TabletPC and desktop computer platforms. The PDA versions were developed in Java, using MIDP 1.0, and were tested on PalmOS and Pocket PCs. Nevertheless, running the prototypes on mobile phones or other devices that have a Java Virtual Machine (JVM) installed should be straightforward. A version with MIDP 2.0 was initially developed, but (known) problems arose with functionalities ill-implemented on the corresponding JVM for PDAs.

As PDAs offer a small screen and limited data input possibilities, challenges increase when designing user centred applications for such platforms. Focused, simple user interface were designed, where data input is facilitated as much as possible. Lists and drop box menus are available for use (upon therapist decision) when filling or creating an artefact, reducing greatly the need to write data on the application. Lists length were condensed and contextualized.

5.1 ScoNOTES – Annotating

Using ScoNOTES the therapist is able to gather information and comment on previously gathered one. From the working patient record, the therapist selects ScoNOTES and simply writes text. The resulting annotations are registered with the creation dates and become automatically accessible from the patient's record. This necessarily simple process is particularly adequate for consultation settings. In this context, the default free-text entry is set to the cursive writing alternative (bitmap).

On an off-consultation setting, ScoNOTES enables the (re)organization of annotations and the creation of more detailed and structured ones. These, in fact, may provide a source of meta-information for every sort of data or artefact that exists in the therapy process. Keywords can be included, themes are reinforced and text entry fields are preferably character-based. New annotations are associated with patients as well, but can be further correlated with every other artefact, including other annotations.

5.2 ScoFORMS – Creating Artefacts

ScoFORMS includes three flavours of the same tool: ScoQUE, ScoTIVITY and ScoTHOUGHTS. Each provides therapists a way to create a specific form used in CBT, namely questionnaires, activity registering artefacts and thought record forms.

ScoQUE. ScoQUE allows therapists to fetch, change or create questions, and build questionnaires adequate to specific patients and pathologies (Fig. 3 shows an example). A pool of standard questions is always available, as well as all those previously created by the therapist. Different navigation arrangements (e.g. theme, severity) and direct keyword search are supported for question selection. The sequence of questions, or its repetition, is decided when constructing a questionnaire. Standard and stereotyped questionnaires are kept and available for reuse.

Each question has associated an answering type (e.g. keyword, text), a set of help topics and a default interaction element (e.g. gauge, text-box). For each question on a particular questionnaire, the therapist is allowed to change the interaction element or combine it with alternative ones, from a set of compatible options. For example, a drop-box instead of a free-text entry or in conjunction with it, as a hint for the patient, is a common choice. Also, the presence or absence of one or more help topics can be adjusted, as well as the total amount of hints and help topics available for a questionnaire. Overall, the therapist is able to customize the questionnaire and the questionnaire's user interface, adapting it to the patient's capabilities.

Figure 2, on the left, shows the ScoQUE interface for selecting questions for a questionnaire. Previously, the questions were inserted into the system, using another interface. At the center, the therapist is choosing a particular answering type, namely a multiple choice. The following step is naturally to define the selectable answers.

Finally, the therapist may also decide if and at what points the patient should get synopsis, score feedback, warnings, incentive words or congratulations. The way these are presented and combined with the patient answers is also configurable. For example, after a task completion, an incentive may appear as a sentence or a graphic depicting the patient's. The application’s behaviour can also be configured to act according to the patients’ choices, detecting whether the patient is answering the questions automatically or if he concentrated in his task. Detection mechanisms range

from the amount of words used in an answer, to the time spent to answer it. For instance, at the right side of fig. 2, the therapist decided that the resulting form will popup a message when the patient selects option 4 for the question presented in the other screenshots of the figure.

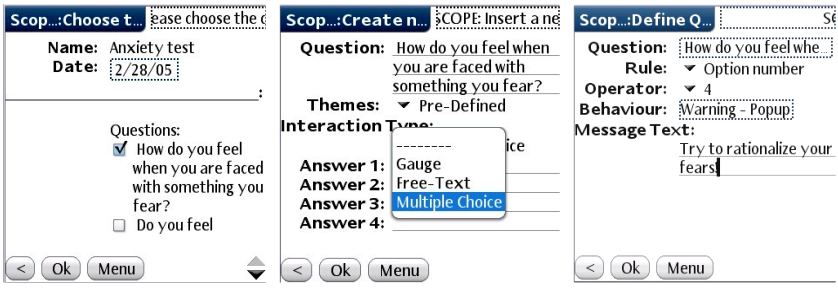


Fig. 2. Building questionnaires with ScoFORMS

ScoTIVITY and ScoTHOUGHTS. These tools allow the creation of artefacts for registering activities and activity plans and thoughts. In general, these flavours are a subset of ScoQUE, as the units (questions) are usually centred on text-entries, within a simple pre-defined structure (e.g. a classification of thought plus a free-text entry). Nonetheless, ScoTIVITY introduces alarms as an activity reminder or planner, and ScoTHOUGHTS assumes units as templates, i.e. a thoughts-form is often a single unit that will be instantiated every time the patient registers a thought. Therapists may also configure help, reports and rewards. A common criterion for rewards is the number of registered thoughts.

5.3 ScoTHERAPY – Patient's Tool

ScoTHERAPY is the tool that materializes the artefacts to be filled by the patients. Its primary goal is to provide a way to quickly answer questionnaires, register thoughts and activities, or plan activities, as part of a therapeutic process. Overall, it is driven by a simple, easy to use interface that can provide access to aids, such as supportive hints, choice lists, help, reporting and rewarding. These aids can be disabled on face-to-face setting scenarios, allowing the therapist to actively intervene on the task fulfilment, or enabled on the patient individual scenarios. On the latter, it is up to the therapist to define when, where and what kind of aids are available. For example, after registering three positive thoughts the patient may be presented with a congratulations message, or, at a later therapy stage, the patient may only have access to hint choice-lists on five of the most complex tasks. Another common usage is to pop a hint when a patient is taking to much time to answer a question.

Figure 3 shows an activity registry on the left and the corresponding help screen on the right. If allowed by the therapist, the patient may, in future activity planning, access activities' specifications that he/she entered on a previous activity filling task. For example, consider a user that on a specific day plans a new activity like "Go out to lunch", writing it down on a text-box. Consider also that the therapist allowed that

patient to access a dropdown-box of activity hints. Then, on the next activity planning task, the patient may find "Go out to lunch" on that dropdown-box. This, naturally, reduces the time and effort spent in this task.

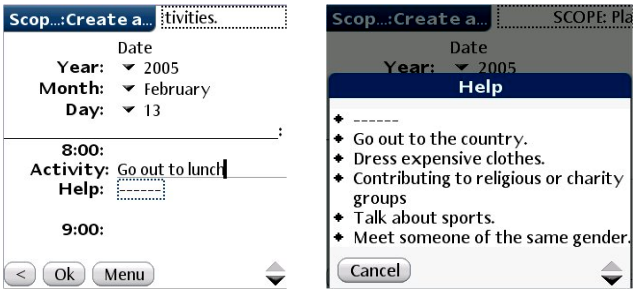


Fig. 3. Filling activities with ScoTHERAPY

The artefacts handled by ScoTHERAPY are transferred between therapist and patient devices as needed. Therapists usually export the forms specifications, eventually containing data from a previous shared-scenario session. Therapists import the patients' filled data, eventually containing new items in choice lists (e.g. recurring activities). Currently the transfer is done using Bluetooth and PalmOS synchronization tools, through a PC or TabletPC. Alternatively ScoTHERAPY can print results when inserted in a more classic process.

5.4 ScopALYSIS – Analysis

ScopALYSIS provides a set of components for the analysis of patient data. Its capabilities range from simple scoring analysis on a questionnaire, usually applied during consultation, to more complex ones, more adequate to after sessions. Rearranging the scoring criteria (e.g. based on questions relevance) or determining score evolution over multiple questionnaires are common usage examples. On thought and activity records ScopALYSIS is able to find recurrent themes and keywords or determine word frequency (Fig. 4). Its corpus can also be extended to annotations, thus building on meta-information introduced by the therapist.

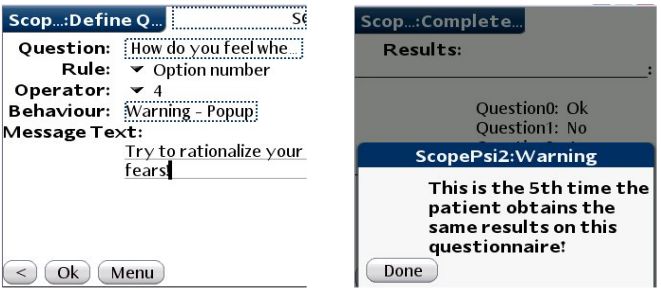


Fig. 4. Defining warning rules with ScopALYSIS

Visualization of the analysis results can be done through tables or simple graphics, as selected by the therapist. On larger device' platforms, diagrammatic (graph based) presentations are available. Filtering of results is also possible.

ScopALYSIS is articulated with all the remaining SCOPE tools. With ScoTHERAPY it provides a way to explicitly work the patient's data. A limited version is integrated with the patients' view of ScoTHERAPY, mostly as a presentation component. To ScoFORMS, ScopALYSIS provides the components to define rewards including reward triggers (metrics and targets) and presentations. Finally, ScopALYSIS is also connected with ScoNOTES since the therapist may wish to comment on analysis results that can be kept along with patients' data.

6 Design and Evaluation

This project followed a contextual design approach [2]. During early stage design, interviews and meetings with different psychologists were made. Documentation and videos [6] describing the therapy and showing real therapy examples were thoroughly studied. Information flows, activities, artefacts, physical settings and cultural issues along with major breaks (complaints, etc.) were identified, leading to the construction of two low-fidelity prototypes.

6.1 Early Stage Evaluation

The two low-fidelity prototypes were used in early stage evaluation. One of the prototypes mimics a PDA and the other a TabletPC. They are composed of a wooden frame where cards, representing screens, can be replaced. The screens are sketches of the applications later refined and implemented on the software prototypes. Card-sets representing the patient and the therapist tools were created for PDA and TabletPC and refined in consequence of successive evaluation sessions.

Low-fidelity prototypes of both platforms were evaluated under a Wizard of Oz approach. Three psychologists, two of them with clinical know-how and 10 other individuals with different education backgrounds, assessed the prototypes. The first group provided a more thorough, domain oriented, evaluation. The latter group was focused on sequence and screen arrangements, as well as on the use of such artefacts during interview conversations. The therapy original paper forms were also provided and procedures explained.

Results were quite encouraging. Psychologists specially welcomed the ability to exchange forms with the patient in a digital format, still maintaining the facility to share and collaborate in the filling process. The customization of forms was particularly well received, as well as the disappearance of breakdowns such as interruptions to get printed forms and fetch previous session results, and editing difficulties during the co-filling process. For the therapist components, the TabletPC versions were preferred due to the small screen size of PDAs when visualizing large amounts of information and introducing data. The psychologists' evaluation of the patients' application was also positive although some rearrangements were suggested for ease of use. The requirement of customizing the patient reward subsystem emerged from this design phase.

The non-expert group, particularly applied to the patients' tools, also triggered the adjustment of some components. Multiple choice questionnaires were very easy to fill

whereas free-text forms raised some usage concerns. However, when confronted with the paper original version, users clearly chose the prototypes, particularly the PDA one. In simulated therapy sessions the prototypes were not found more intrusive than paper.

6.2 Software Prototypes Evaluation

The software prototypes were tested with the same group of users that tested the low-fidelity ones, plus computer engineering students and a group of 3 users with no experience in using handheld devices. The main focus of these last two groups was directed to interaction, usage and design. The patient application was actually used during a couple of days to perform specific form filling tasks. Measurements were made and usability questionnaires responded during each session. Overall the three groups found the software prototypes very promising.

Apart from the original expert group, another experienced clinical psychotherapist has evaluated the improved versions of the software prototypes. He contributed with suggestions of new evolutions and recent therapy practices that can extend SCOPE's coverage. The introduction of a positive/negative classifying field on thought records emerged from this evaluation. Moreover, the ability to provide a hierarchically organized selection of thoughts and emotions was also suggested. Interestingly, only minor changes on ScoFORMS were required to accommodate these needs.

Further results on the evaluation of the software prototypes can be seen in [18].

6.3 Building Existing Applications

To evaluate the flexibility of the tools, particularly of the ScoFORMS family and ScoTherapy, existing therapy applications were partially created using the former and tested using the latter. A TabletPC-based version of the form building tool [18] was also used in order to facilitate the creation phase. The PDA version was basically used for refinement, although most of the building features were available in both platforms.

A set of applications paper-based or already PDA-based were selected for anxiety, depression, pain and weight disorders. Therapists, sometimes requiring the help of the development team, were requested to build those applications. The applications were successfully created as documented in [18]. As a consequence of these evaluation process some refinement was done on both tools.

6.4 Evaluating in Clinical Settings

Currently, a psychotherapy research group is using SCOPE for the treatment of recovering cancer patients and pain associated disorders. Artefacts and forms were integrated by the therapists into specific applications and are being used by actual patients [17]. Conclusive results are yet to be obtained as the therapy process takes considerable time.

7 Conclusions

The use of technology in health care is evolving quite rapidly. Psychotherapy, in particular, is one of the most promising areas for technology application, especially for its hand-held and collaborative requirements. However, even if some applications

exist, they are rigid, focussed only on the therapy itself and not considering the actors specificities, not integrated and often missing the context it should serve. The work described in this paper aims to overcome these difficulties and support psychotherapy through all the process.

The paper focuses on the PDA versions of a set of components that cover therapist annotation, prescription and analysis tasks. The first one addresses fast and structured annotation. A second central component provides the therapist with the ability to define the artefacts used in the patient tasks and the characteristics of his "substitute" as a patient aid, i.e., the therapist is able to define the contents and the look and feel of the forms to be filled by the patient and to establish what, when and how the aids will be available or presented to the patient.

The paper also describes the tool that enables the patient to fill the artefacts and the therapist to browse the patient's data. The last component pertains to the analysis of the patient data. Overall these tools cover most of the therapists' activities on a therapeutic process, including the assisted filling of forms, wherever they take place, proactively reacting to the patients' behaviour and evolution. This mobility and ubiquity and the ability given to the therapist to create and adjust active artefacts, enhancing psychotherapy, are the major contributions of the SCOPE project.

The work done so far has been validated, in its various stages, by a strict collaboration with several professionals and researchers in psychotherapy. They have consistently provided useful input and opinions, guaranteeing the developed prototypes' value, and providing coherent directions for evolution. A quite interesting observation is that when new specific therapies were suggested, for example weight and pain control, the creation of the corresponding artefacts is easily accomplished with ScoFORMS and its usage easily handled with ScoTHERAPY.

Finally, new functionalities are planned to be included in SCOPE. For example: voice interaction with the mobile device; the use of different media like videos or audio files explaining certain procedures; alarms that alert patients or even their therapists of specific situations.

Currently, our efforts are also being directed to a group version of the described tools [3]. These include a managing tool which allows therapists to engage on group therapy sessions, with patients that might be in the same room or distributed through several locations. Furthermore, we are also including a special version for in-exposure therapy sessions, where therapist and patients will be able to engage on group sessions outside the office.

Regarding technology, a wider range of applications, other than CBT, are also on their way. Some patients may need to monitor some symptoms during their day-to-day life. Therefore, the use of automatic monitoring devices, which can control heart rate beats, temperature, etc., communicating directly with the mobile device and triggering alerts when necessary, will also be incorporated into the system.

Acknowledgements

This work was partially funded by FCT (Fundação para a Ciência e Tecnologia), through the SCOPE (Supporting Cognition Outlines on Psychological Evaluation - POSI/SRI/44247/2002) and JoinTS (Joint Psychological Therapy Support) projects.

References

1. Andersson, G., Kaldo, V.: Internet-Based Cognitive Behavioural Therapy for Tinnitius. Journal of Clinical Psychology 60, 171–178 (2004)
2. Beyer, H., Holtzblatt, K.: Contextual Design: A Customer-Centered Approach to Systems Design. Academic Press, San Francisco, CA, USA (1998)
3. Carriço, L., Sá, M., Duarte, L., Carvalho, J.: Managing Group Therapy through Multiple Devices. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4553, pp. 427–436. Springer, Heidelberg (2007)
4. Carriço, L., Reis, J., Duarte, C., Henriques, A.: Guimarães, N.: Scope supporting cognition outlines on psychological evaluation. Technical Report POSI SRI/44247/2002 - Project Proposal, LaSIGE-FCUL/PsiSaude (2002)
5. Das, A.K.: Computers in Psychiatry: A Review of Past Programs and an Analysis of Historical Trends. Psychiatry Quarterly 79(4) (2002) Winter
6. Davidson, J., Persons, J.B., Tompkins, M.A.: Cognitive-behavior therapy for depression. In: Psychotherapy Video Tape Series, American Psychology Association (2000)
7. Garrard, C.S.: Human-computer interactions: can computers improve the way doctors work? Schweitz Med Wochenschr 130, 1557–1563 (2000)
8. Grasso, M.A.: Clinical Applications of Hand Held Computing. In: Proceedings of the 17th IEEE Symposium on Computer Based Medical Systems (CBMS), IEEE Computer Society Press, Los Alamitos (2004)
9. Gega, L., Marks, I., Mataix-Cols, D.: Computer-Aided CBT Self-Help for Anxiety and Depressive Disorders: Experience of a London Clinic and Future Directions. Journal of Clinical Psychology 60 (2004)
10. Herman, S., Koran, L.: In vivo measurement of obsessive-compulsive disorder symptoms using palmtop computers. Computers in Human Behaviour 14(3), 449–462 (1998)
11. Luff, P., Heath, C.: Mobility in Collaboration. In: CSCW 1998. Proceedings of Computer Supported Cooperative Work, ACM Press, New York (1998)
12. Mahoney, M.: Constructive Psychotherapy. The Guilford Press, New York (2003)
13. Newman, M.G.: Technology in psychotherapy: an introduction. Journal of Clinical Psychology 60(2) (2004)
14. Otto, M.L., Pollack, M.H., Maki, K.M.: Empirically supported treatments for panic disorder: costs, benefits and stepped care. Journal Consult Clinical Psychology 68 (4) (2000)
15. Proudfoot, J.: Computer-based treatment for anxiety and depression: is it feasible? Is it effective? Neuroscience and Biobehavioral Reviews 28 (2004)
16. Przeworski, A., Newman, M.G.: Palmtop computer-assisted group therapy for social phobia. Journal of Clinical Psychology 60(2), 179–188 (2004)
17. Reis, J., Travado, L., Lucas, H., Carriço, L., Sá, M.: Cognitive-behavioral therapy with an oncological patient via handheld computing. In: Proceedings of 8th world congress of psycho-oncology,Suppl. Meeting Abstract, Venice, Italy, pp. 905–906. Wiley, Chichester (2006)
18. Sá, M., Carriço, L., Antunes, P.: Ubiquitous Psychotherapy. IEEE Pervasive Computing 6(1), 20–27 (2007), Special Issue on Healthcare, IEEE Press
19. Tate, D., Zabinski, M.: Computer and Internet Applications for Psychological Treatment: Update for Clinicians. Journal of Clinical Psychology 60 (2004)
20. Wright, J.H., Wright, A.: Computer-assisted psychotherapy. Journal of Psychotherapy Practice Research 6, 315–319 (1997)

Author Index