Silvana Castano
Panos Vassiliadis
Laks V. S. Lakshmanan
Mong Li Lee (Eds.)

LNCS 7518

Advances in Conceptual Modeling

ER 2012 Workshops: CMS, ECDM-NoCoDA, MoDIC
MORE-BI, RIGiM, SeCoGIS, WISM
Florence, Italy, October 2012
Proceedings

Springer

Lecture Notes in Computer Science 7518

Silvana Castano   Panos Vassiliadis
Laks V. S. Lakshmanan   Mong Li Lee (Eds.)

Advances in Conceptual Modeling

ER 2012 Workshops: CMS, ECDM-NoCoDA, MoDIC,
MORE-BI, RIGiM, SeCoGIS, WISM
Florence, Italy, October 15-18, 2012
Proceedings

Springer

Volume Editors

Silvana Castano
Università degli Studi di Milano
Via Comelico 39
20135, Milano, Italy
E-mail: silvana.castano@unimi.it

Panos Vassiliadis
University of Ioannina
Ioannina, 45110 Greece
E-mail: pvassil@cs.uoi.gr

Laks V. S. Lakshmanan
University of British Columbia
2366 Main Mall
Vancouver, BC, V6T 1Z4 Canada
E-mail: laks@cs.ubc.ca

Mong Li Lee
National University of Singapore
13 Computing Drive
Singapore 117417 Singapore
E-mail: leeml@comp.nus.edu.sg

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-33998-1 e-ISBN 978-3-642-33999-8
DOI 10.1007/978-3-642-33999-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012948191

CR Subject Classification (1998): I.6.5, H.4.1-3, H.2.7-8, H.3.3-4, I.2.4, I.2.10, D.2, H.5.3, C.2.4

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the proceedings of the workshops associated with the 31st International Conference on Conceptual Modeling (ER 2012) and the papers associated with the Demonstration Session of ER 2012.

Continuing the ER tradition, the ER 2012 workshops provided researchers, students, and industry professionals with a forum to present and discuss emerging, cutting-edge topics related to conceptual modeling and its applications. After a call for workshop proposals, the following seven workshops were selected and organized out of 12 high-quality submissions:

- CMS, Third International Workshop on Conceptual Modelling of Services
- ECDM-NoCoDa, International Workshop on Evolution and Change in Data Management (ECDM) and Non-Conventional Data Access (NoCoDA)
- MoDIC, International Workshop on Modeling for Data-Intensive Computing
- MORE-BI, Second International Workshop on Modeling and Reasoning for Business Intelligence
- RIGiM, 4th International Workshop on Requirements, Intentions and Goals in Conceptual Modeling
- SeCoGIS, 6th International Workshop on Semantic and Conceptual Issues in Geographic Information Systems
- WISM, 9th International Workshop on Web Information Systems Modeling

The workshops cover different conceptual modeling topics, from requirements, goal and service modeling, to evolution and change management, to non-conventional data access, and they span a wide range of domains including Web information systems, geographical information systems, business intelligence, data-intensive computing.

These seven workshops received a total amount of 84 submissions. Following the rule of the ER workshops, the respective workshop Program Committees carried out peer reviews and accepted a total number of 32 papers with an acceptance rate of 38%. We also invited speakers who significantly enhanced the perspectives and quality of the ER 2012 workshops. For the demonstration session, six papers were selected with the aim of demonstrating new and innovative applications.

Setting up workshops and demonstrations required a lot of effort and involved many people. We would like to thank the workshop organizers, for their invaluable collaboration and their significant effort to provide a successful workshop edition; the workshops Program Committees and external reviewers, for their professional contribution to the paper review that ensured such a high-quality program; the Program Committee of the demonstrations for the diligence in selecting the papers in this volume.

We would also thank the conference Co-chairs, Valeria De Antonellis and Stefano Rizzi, for their continuous support and the help in setting up the final program. A very special thanks to Stefano Montanelli for his hard work in editing this volume.

Finally, we would like to express our sincere appreciation to the authors of all the submitted papers for the high-quality contributions. We rely on their continuous support to keep up the high quality of the ER conference.

July 2012

Silvana Castano
Panos Vassiliadis
Laks V.S. Lakshmanan
Mong Li Lee

Organization

ER 2012 Workshop chairs

Silvana Castano	Università degli Studi di Milano, Italy
Panos Vassiliadis	University of Ioannina, Greece

CMS 2012

Program Chairs

Klaus-Dieter Schewe	Software Competence Center Hagenberg, Austria and Johannes-Kepler University Linz, Austria
Qing Wang	The National University of Australia, Australia

Program Committee

Yamine Aït-Ameur	IRIT/ENSEEIHT, France
Karoly Bosa	Johannes-Kepler University Linz, Austria
Schahram Dustdar	Vienna University of Technology, Austria
Daniele Gianni	European Space Agency, The Netherlands
Paul Johanneson	University of Stockholm, Sweden
Markus Kirchberg	Visa Research, Singapore
Werner Kurschl	University of Applied Sciences Upper Austria, Austria
Hui Ma	Victoria University of Wellington, New Zealand
Dana Petcu	West University of Timisoara, Romania
Bernhard Thalheim	Christian Albrechts University Kiel, Germany
Yan Zhu	Southwest Jiaotong University Chengdu, China
Thomas Ziebermayr	Software Competence Center Hagenberg, Austria

ECDM – NoCoDa 2012

Program Chairs

Fabio Grandi	Università di Bologna, Italy
Giorgio Orsi	University of Oxford, UK
Letizia Tanca	Politecnico di Milano, Italy
Riccardo Torlone	Università Roma Tre, Italy

Program Committee

(ECDM Track)

Alessandro Artale	Free University of Bolzano-Bozen, Italy
Ladjel Bellatreche	ENSMA, France
Sourav Bhowmick	Nanyang Technical University, Singapore
Carlo Combi	Università di Verona, Italy
Carlo Curino	MIT, USA
Tudor Dumitras	Symantec Research Labs, USA
Curtis Dyreson	Utah State University, USA
Shashi Gadia	Iowa State University, USA
Renata Galante	Federal University of Rio Grande do Sul, Brazil
Georg Krempl	Otto von Guericke University Magdeburg, Germany
Jorge Lloret	University of Zaragoza, Spain
Federica Mandreoli	Università di Modena and Reggio Emilia, Italy
Torben Bach Pedersen	Aalborg University, Denmark
John Roddick	Flinders University, Australia
Nandlal Sarda	IIT Bombay, India
Paolo Terenziani	Università di Torino, Italy
Carlo Zaniolo	UCLA, USA

(NoCoDa Track)

Leopoldo Bertossi	Carleton University, Canada
Francois Bry	Ludwig Maximilians Universität München, Germany
Andrea Calì	Birkbeck University of London, UK
Diego Calvanese	Università di Bolzano, Italy
Paolo Ciaccia	Università di Bologna, Italy
George Fletcher	Technische Universiteit Eindhoven, The Netherlands
Georg Gottlob	University of Oxford, UK
Francesco Guerra	Università di Modena e Reggio Emilia, Italy
Georgia Koutrika	IBM Almaden, USA
Marco Manna	Università della Calabria, Italy
Ioana Manolescu	INRIA/LRI, France
Davide Martinenghi	Politecnico di Milano, Italy
Wilfred Ng	University of Hong Kong, SAR China
Dan Olteanu	University of Oxford, UK
Jan Paredaens	Universiteit Antwerpen, Belgium
Andreas Pieris	University of Oxford, UK
Evaggelia Pitoura	University of Ioannina, Greece
Mike Rosner	University of Malta, Malta

Timos Sellis	Research Center “Athena” and NTUA, Greece
Pierre Senellart	Télécom ParisTech
Stijn Vansummeren	Université Libre de Bruxelles, Belgium
Yannis Velegrakis	Università degli studi di Trento, Italy

MoDIC 2012

MoDIC 2012 was organized within the framework of the following projects: MESOLAP (TIN2010-14860) and SERENIDAD (PEII-11-0327-7035) projects from the Spanish Ministry of Education and the Junta de Comunidades de Castilla La Mancha respectively.

Program Chairs

David Gil	University of Alicante, Spain
Il-Yeol Song	Drexel University, USA
Juan Trujillo	University of Alicante, Spain

Program Committee

Michael Blaha	Yahoo Inc., USA
Rafael Berlanga	Universitat Jaume I, Spain
Gennaro Cordasco	Università di Salerno, Italy
Alfredo Cuzzocrea	University of Calabria, Italy
Gill Dobbie	University of Auckland, New Zealand
Eduardo Fernández	University of Castilla-La Mancha, Spain
Matteo Golfarelli	University of Bologna, Italy
Nectarios Koziris	Technical University of Athens, Greece
Jiexun Li	Drexel University, Philadelphia, USA
Alexander Loeser	Universität Berlin, Germany
Héctor Llorens	University of Alicante, Spain
Antoni Olivé	Universitat Politécnica de Catalunya, Spain
Jeff Parsons	Memorial University of Newfoundland, Canada
Oscar Pastor	Universitat Politècnica de València, Spain
Mario Piattini	University of Castilla-La Mancha, Spain
Sudha Ram	University of Arizona, USA
Colette Roland	University of Paris 1-Panthéon Sorbonne, France
Keng Siau	University of Nebraska-Lincoln, USA
Alkis Simitsis	Hewlett-Packard Co, California, USA
Julia Stovanovich	University of Pennsylvania, USA
Alejandro Vaisman	Université Libre de Bruxelles, Belgium
Sergei Vassilvitskii	Yahoo Inc., USA

MORE-BI 2012

Program Chairs

Ivan J. Jureta	University of Namur, Belgium
Stéphane Faulkner	University of Namur, Belgium
Esteban Zimányi	Université Libre de Bruxelles, Belgium

Program Committee

Alberto Abelló	Universitat Politècnica de Catalunya, Spain
Daniele Barone	University of Toronto, Canada
Ladjel Bellatreche	Ecole Nationale Supérieure de Mécanique et d'Aérotechnique, France
Sandro Bimonte	Irstea, Clermont-Ferrand, France
Farid Cerbah	Dassault Aviation, France
Dalila Chiadmi	Ecole Mohammadia d'Ingénieurs, Morocco
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria, Italy
Olivier Corby	INRIA, France
Marin Dimitrov	Ontotext, Bulgaria
Neil Ernst	University of British Columbia, Canada
Cécile Favre	Université Lyon 2, France
Octavio Glorio	University of Alicante, Spain
Matteo Golfarelli	University of Bologna, Italy
Gregor Hackenbroich	SAP, Germany
Dimitris Karagiannis	University of Vienna, Austria
Alexei Lapouchnian	University of Trento, Italy
Sotirios Liaskos	York University, UK
Isabelle Linden	University of Namur, Belgium
Patrick Marcel	Université François Rabelais de Tours, France
Jose-Norberto Mazón	University of Alicante, Spain
Jeffrey Parsons	Memorial University of Newfoundland, Canada
Anna Perini	Fondazione Bruno Kessler, Italy
Stefano Rizzi	University of Bologna, Italy
Monique Snoeck	Katholieke Universiteit Leuven, Belgium
Catherine Roussey	Irstea, Clermont-Ferrand, France
Thodoros Topaloglou	University of Toronto, Canada
Juan-Carlos Trujillo Mondéjar	University of Alicante, Spain
Robert Wrembel	Poznań University of Technology, Poland

RIGiM 2012

Program Chairs

Colette Rolland	Université Paris 1 Panthéon - Sorbonne, France
Jennifer Horkoff	University of Toronto, Canada
Eric Yu	University of Toronto, Canada
Camille Salinesi	Université Paris 1 Panthéon - Sorbonne, France
Jaelson Castro	Universidade Federal de Pernambuco, Brazil

Program Committee

Raian Ali	Bournemouth University, UK
Thomas Alspaugh	University of California, Irvine, USA
Daniel Amyot	University of Ottawa, Canada
Mikio Aoyoma	Nanzan University, Japan
Ian Alexander	Scenario Plus, UK
Daniel Berry	University of Waterloo, Canada
Luiz Cysneiros	York University, Canada
Fabiano Dalpiaz	Trento University, Italy
Vincenzo Gervasi	University of Pisa, Italy
Aditya K. Ghose	University of Wollongong, Australia
Paolo Giogini	University of Trento, Italy
Renata Guizzardi	Universidade Federal do Espírito Santo (UFES), Brazil
Patrick Heymans	University of Namur, Belgium
Zhi Jin	Chinese Academy of Sciences, China
Haruhiko Kaiya	Shinshu University, Japan
Aneesh Krishna	Curtin University, Australia
Régine Laleau	Université Paris XII, France
Axel van Lamsweerde	Université Catholique de Louvain, Belgium
Alexei Lapouchnian	University of Trento, Italy
Julio Leite	Pontificia Universidade Catolica, Brazil
Emmanuel Letier	University College of London, UK
Sotirios Liaskos	York University, Canada
Lin Liu	Tsinghua University, China
Peri Loucopoulos	University of Manchester, UK
Andreas Opdahl	University of Bergen, Norway
Anna Perini	FBK - Fondazione Bruno Kessler, Italy
Barbara Pernici	Politecnico di Milano, Italy
Yves Pigneur	HEC, Lausanne, Suisse
Jolita Ralyte	University of Geneva, Switzerland
Motoshi Saeki	Tokyo Institute of Technology, Japan
Pnina Soffer	University of Haifa, Israel
Sam Supakkul	Keane, An NTT DATA Company, USA

Angelo Susi	FBK - Fondazione Bruno Kessler, Italy
Roel Wieringa	University of Twente, The Netherlands
Carson Woo	University of British Columbia, Canada

SeCoGIS 2012

Program Chairs

Eliseo Clementini	University of L'Aquila, Italy
Esteban Zimanyi	Université Libre de Bruxelles, Belgium

Program Committee

Alia I. Abdelmoty	Cardiff University, UK
Phil Bartie	University of Edinburgh, UK
Claudia Bauzer Medeiros	University of Campinas, Brazil
Yvan Bédard	Université Laval, Canada
David Bennett	University of Iowa, USA
Michela Bertolotto	University College Dublin, Ireland
Roland Billen	Université de Liège, Belgium
Patrice Boursier	University of La Rochelle, France
Jean Brodeur	National Resources Canada, Canada
Bénédicte Bucher	Institut Géographique National, France
Adrijana Car	German University of Technology, Oman
Christophe Claramunt	Naval Academy Research Institute, France
Maria Luisa Damiani	University of Milan, Italy
Clodoveu Davis	Federal University of Minas Gerais, Brazil
Gilles Falquet	University of Geneva, Switzerland
Fernando Ferri	Istituto di Ricerche sulla Popolazione e le Politiche Sociali, Italy
Paolo Fogliaroni	University of Bremen, Germany
Andrew Frank	Technical University of Vienna, Austria
Bo Huang	The Chinese University of Hong Kong, China
Marinos Kavouras	National Technical University of Athens, Greece
Ki-Joune Li	Pusan National University, South Korea
Thérèse Libourel	Université de Montpellier II, France
Jugurta Lisboa Filho	Universidade Federal de Viçosa, Brazil
Miguel R. Luaces	Universidade da Coruña, Spain
Jose Macedo	Federal University of Ceara, Brazil
Pedro Rafael Muro Medrano	University of Zaragoza, Spain
Peter van Oosterom	Delft University of Technology, The Netherlands
Dimitris Papadias	Hong Kong University of Science and Technology, Hong Kong

Dieter Pfoser	Institute for the Management of Information Systems, Greece
Ricardo Rodrigues Ciferri	Universidade Federal de São Carlos, Brazil
Andrea Rodriguez Tastets	University of Concepción, Chile
Markus Schneider	University of Florida, USA
Sylvie Servigne-Martin	INSA de Lyon, France
Shashi Shekhar	University of Minnesota, USA
Spiros Skiadopoulos	University of the Peloponnese, Greece
Emmanuel Stefanakis	Harokopio University of Athens, Greece
Kathleen Stewart Hornsby	University of Iowa, USA
Kerry Taylor	CISRO, Australia
Sabine Timpf	University of Augsburg, Germany
Antonio Miguel Vieira Monteiro	INPE, Brazil
Nancy Wiegand	University of Wisconsin, USA
Stephan Winter	University of Melbourne, Australia

External Referees

Zhe Jiang
Javier Lacasta
Javier Nogueras-Iso

WISM 2012

Program Chairs

Flavius Frasincar	Erasmus University Rotterdam, The Netherlands
Geert-Jan Houben	Delft University of Technology, The Netherlands
Philippe Thiran	Namur University, Belgium

Program Committee

Syed Sibte Raza Abidi	Dalhousie University, Canada
Djamal Benslimane	University of Lyon 1, France
Marco Brambilla	Politecnico di Milano, Italy
Sven Casteleyn	Universidad Politecnica de Valencia, Spain
Richard Chbeir	Bourgogne University, France
Jose Palazzo Moreira de Oliveira	UFRGS, Brazil
Olga De Troyer	Vrije Universiteit Brussel, Belgium
Roberto De Virgilio	Università di Roma Tre, Italy
Oscar Diaz	University of the Basque Country, Spain

Flavius Frasincar	Erasmus University Rotterdam, The Netherlands
Irene Garrigos	Universidad de Alicante, Spain
Hyoil Han	LeMoyne-Owen College, USA
Geert-Jan Houben	Delft University of Technology, The Netherlands
Zakaria Maamar	Zayed University, UAE
Michael Mrissa	University of Lyon 1, France
Moira Norrie	ETH Zurich, Switzerland
Oscar Pastor	Valencia University of Technology, Spain
Dimitris Plexousakis	University of Crete, Greece
Davide Rossi	University of Bologna, Italy
Hajo Reijers	Eindhoven University of Technology, The Netherlands
Klaus-Dieter Schewe	Johannes Kepler University Linz, Austria
Bernhard Thalheim	Christian Albrechts University Kiel, Germany
Philippe Thiran	Namur University, Belgium
Riccardo Torlone	Università di Roma Tre, Italy
Lorna Uden	Staffordshire University, UK
Erik Wilde	UC Berkeley, USA

ER 2012 Demonstrations

Program Chairs

Laks V.S. Lakshmanan	The University of British Columbia, Canada
Mong Li Lee	National University of Singapore, Singapore

Program Committee

Sourav Bhowmick	Nanyang Technological University, Singapore
Francesco Bonchi	Yahoo! Research, Barcelona, Spain
Gillian Dobbie	University of Auckland, New Zealand
Jiaheng Lu	Renmin University of China, China
Amelie Marian	Rutgers University, USA
Divesh Srivastava	AT&T Labs Research, USA
Sergei Vassilivtskii	Yahoo! Research, USA
Cong Yu	Google Research, USA
Jeffrey Yu	City University of Hong Kong, Hong Kong
Haixun Wang	Microsoft Research Asia, China

Table of Contents

CMS 2012 – Third International Workshop on Conceptual Modelling of Services

Session I: Ontologies and Web Applications

Session II: Services and Cloud Computing

ECDM-NoCoDa 2012 – First International Workshop on Evolution and Change in Data Management and on Non Conventional Data Access

Session I: Evolution and Change in Data Management

Session II: Non Conventional Data Access

MoDIC 2012 – First International Workshop on Modeling for Data-Intensive Computing

Session I: Big Data: General Issues and Modeling Approaches

Session II: Ontologies and Conceptual Models

MORE-BI 2012 – Second International Workshop on Modeling and Reasoning for Business Intelligence

RIGiM 2012 – Fourth International Workshop on Requirements, Intentions and Goals in Conceptual Modeling

SeCoGIS 2012 – Sixth International Workshop on Semantic and Conceptual Issues in GIS

Session I: New Frontiers of Spatio-temporal Dimensions

Session II: Semantic Issues

Session III: Conceptual Issues

Session IV: Spatio-temporal Issues

WISM 2012 – Ninth International Workshop on Web Information Systems Modeling

Session I: Web Information Systems Development and Analysis Models

Session II: Web Technologies and Applications

ER 2012 – Demonstrations

Third International Workshop on Conceptual Modelling of Services (CMS 2012)

Preface

The CMS workshop aims at bringing together researchers in the areas of services computing, services science, business process modelling, and conceptual modelling. The emphasis of this workshop is on the intersection of the rather new, fast growing services computing and services science paradigms with the well established conceptual modelling area.

The first CMS workshop was held in November 2010 in Vancouver, Canada in connection with ER 2010. The second CMS workshop was held in September 2011 in Milan, Italy in connection with ICDKE 2011. The call for research papers for CMS 2012 solicitated the following topics:

- Conceptual models for integrated design and delivery of value bundles (i.e. services and physical goods)
- Formal methods for services computing / services science
- Foundation of business process modelling, integration and management
- Modelling languages / techniques for services
- Modelling of semantic services
- Modelling support for service integration – within a single domain, across multiple domains, with legacy applications, etc.
- Personalisation of services
- Quality of service modelling
- Reference models for service-oriented systems
- Semantics of service-oriented systems
- Service composition planning and verification
- Service computing process modelling, transformation and integration
- Service development process modelling
- Service ontologies and ontology alignment
- User interfaces for assisted service modelling

Out of twelve submitted papers to the workshop the international programme committee selected four papers for presentation at the workshop. These papers are included in these proceedings.

In addition, Vincenzo Gervasi from University of Pisa, Italy gave a keynote on “Modeling web applications with Abstract State Machines – refining concepts into technology”. The keynote emphasised that modeling web applications and web services offer interesting challenges, because their conceptual design is so intertwined with the historical accidents of their development, and with limitations and features of the underlying technologies, that capturing the essence of their behaviour in a formal model requires a delicate hand. A good formal model should serve not only as a precise description of the expected behaviour (i.e., a specification), but also as an aid for human understanding and communication. A formal model so complicated that its contents become unaccessible to the reader is of little use for that purpose. The keynote discussed how these challenges have been tackled by using the Abstract State Machines (ASMs) approach. ASMs, with their rich notion of state, support for parallelism and non-determinism, and their familiar notation, offer a fertile ground for experimenting with means of clearly communicating precise semantics to non-specialists. At the same time, it is important not to “abstract reality away”, and thus it is nrecessary to incorporate in an identifiable manner the limits and perks of current technology in the model. In so doing, the keynote highlighted some subtler points of both web applications and web services, and of the ASM method itself.

We would like to thank our keynote speaker Vincenzo Gervasi, all authors, presenters and reviewers for their work that helped intensifying the connection between service-oriented systems and conceptual modelling. We are also grateful to the ER 2012 workshop chairs for giving us the opportunity to use this conference as a platform for the continuation of the CMS workshops.

October 2012

Klaus-Dieter Schewe
Qing Wang

A Rule Based Approach for Mapping Sensor Data to Ontological Models in AAL Environments

Mario Buchmayr[1], Werner Kurschl[2], and Josef Küng[3]

[1] Research Center Hagenberg, 4232 Hagenberg, Austria
Mario.Buchmayr@fh-hagenberg.at
[2] Upper Austria University of Applied Sciences, 4232 Hagenberg, Austria
Werner.Kurschl@fh-hagenberg.at
[3] Institute for Application Oriented Knowledge Processing, 4020 Linz, Austria
jkueng@faw.jku.at

Abstract. Improved sensing technologies and cheap sensor devices facilitate the creation of Ambient Assisted Living (AAL) environments. Whereas the increasing manifoldness of sensing possibilities helps to gain detailed and precise information about the environment, the task of dynamically mapping data from different sensor sources to a processable data model becomes more and more complex. Especially in AAL environments which build upon different sensors and sensor networks, the integration of distinct data sources becomes an issue. Most systems cope with this issue by requiring hand written adapters which encapsulate the device communication as well as the data mapping logic. Within this paper we tackle the problem of mapping perceived sensor data to a formal (ontology-based) model with a semi-automated approach. We split up the mapping in two separate parts: *(i)* a protocol specific adapter which encapsulates the communication with the sensor device and *(ii)* a mapping description. The mapping description is specific for each sensor type and defines how the sensed data is mapped to the data model. In addition the mapping description can be used to enrich the data model with additional information, like time and space.

Keywords: Ambient Assisted Living (AAL), Data Model, Data Mapping, Ontology.

1 Introduction

The increasing number of sensor devices for home automation as well as the decreasing prices for such devices facilitate their practical application in smart home environments. The ongoing development on the sensor market, especially the availability of affordable and energy efficient wireless sensors, allows to equip existing or new buildings with different sensors. Partially, such of the shelf sensors can be used in *Ambient Assisted Living (AAL)* environments [3], like residential care homes, foster homes or private homes. The main purpose of AAL is to

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 3–12, 2012.

support elderly or handicapped people in their daily living activities [5] and enable them to stay longer in their familiar environment instead of moving to a nursing home or becoming dependent on home care. To achieve this goal a reliable and proper detection of activities and situations must be provided by an AAL system for two reasons: *(i)* to detect dangerous or life critical situations for the resident and *(ii)* to support and assist the resident in his *Activities of Daily Living (ADL)* or remind him of important tasks (switching off the stove, taking prescribed medicine).

To detect situations or activities a *Data Fusion Component* [8], for merging information from different data sources (i. e. Smart Home sensor values) for decision making, is inevitable. The reasoning about situations is typically done on higher levels of information processing, where basically two approaches are applied for situation detection: *(i)* supervised learning and *(ii)* unsupervised learning by applying knowledge driven methods. Depending on the purpose of the AAL system both approaches have their benefits and drawbacks. In case of extensibility knowledge driven approaches are more suitable than supervised learning, because they offer the possibility to add additional or new data sources (in our case sensor devices) to an existing system without re-training the system. The additional provided knowledge can be considered by adapting existing rules or adding new rules to process the additionally available information. Therefore, it is possible to change a running system (attach or detach new sensors) during runtime and process the sensed data without any change in the existing reasoning infrastructure.

2 Related Work

A variety of AAL environments, especially frameworks like SOPRANO [9], OASIS [4] and PERSONA [1] follow a knowledge driven approach. These frameworks use ontologies to describe, store and reason about the sensed environmental knowledge. SOPRANO, OASIS as well as PERSONA are based on the OSGi Service Platform [11] which provides an infrastructure for service development, like an execution environment, a life cycle management, a service registry and much more. An analysis of the named frameworks showed, that the data mapping is done in a data layer where hand written OSGi adapters are used for mapping raw sensor data to ontological objects.

We assume that it must be possible to ease the data mapping of raw sensor data to ontological objects by a knowledge driven approach. It should be possible to define data mapping constraints for a set of sensors once and automatically map all incoming data from these sensors to ontological objects.

Within this paper we describe an approach for semi-automated data mapping of sensor data to ontological objects in service environments. To demonstrate its feasibility we implemented a prototype which will be discussed in more detail in the following sections.

3 The Data Mapping Issue

Every AAL system faces the issue of how to map sensed data values to data types or objects for further processing. Depending on the used sensor infrastructure and the main purpose of the AAL system, sensor data is either automatically perceived (push principle) or must be queried in temporal intervals from different devices (pull principle). Another problem are the transmission protocols used to communicate with the devices. Most home automation sensors provide a stream of binary information which is transmitted using either a standardized protocol, like TCP/IP, Bluetooth (IEEE 802.15.4) or ZigBee (IEEE 802.15.4), or a proprietary protocol, like C-Bus or ANT. To abstract from these device and communication specific issues, typically one or more abstractions layers are introduced. The benefit gained by using abstraction layers is, that the complexity of device communication can be hidden behind well defined interfaces.

In AAL systems designed and implemented for a fixed set of use cases the mapping of sensor data is trivial. The number and type of used sensor devices can be planned in advance. Afterwards, the sensor locations and communication protocols are settled and will not (can not) be changed anymore. Unfortunately AAL systems require to be much more flexible, because real life behaviour deviates from laboratory simulation and it is essential to gather data from real in-home installations for machine learning and further analysis [7]. Therefore, it must be possible to adapt or change sensor devices after the system has been installed. In addition it is not always possible to install the desired sensors (costs, no retrofittable buildings) or the used sensors might still be under development and are not available yet.

Another issue is the deployment and updatability of the system. Especially, when it is required to update or deploy the system during runtime. To tackle these challenge a service-oriented architecture based upon a service infrastructure (like OSGi) can be used. Besides, the service infrastructure a component for mapping sensed data to data objects would be desirable. Ideally the mapping logic should be defined outside the mapping component to allow to change the mapping behaviour independently from the system. Therefore, a formal model which allows to specify mapping relations among sensor and data values is required.

4 Mapping Approach

Our mapping approach introduces an ontology which serves as generic model for sensed data. This ontology, called SENIOR Core Ontology (sen_core), defines basic relations among sensor objects and data objects. Based on sen_core we derived the SENIOR Sense Ontology (sen_sense), a small ontology specifically designed for an AAL home environment. Sen_sense defines concrete sensors, like a motion sensor or a switch sensor which are installed in a specific building. Figure 1 gives a short overview of our mapping approach and the related components. For each sensor type an adapter (1) must be available to encapsulate the device specific communication. These adapters are registered to a generic data processing component called *Data Mapping Service (DMS)*. The DMS creates a generic model (2+3) related to sen_core and the sensed values. Based on sen_sense and sen_core rules (4) can be defined, which describe how the data from the sen_core model is mapped to a specific sen_sense model. This is done by an *Inference Machine* which will create the *Mapped Model* (5). This approach offers the benefit, that only the mapping rule files need to be created or adapted if additional sensors are added [1]. In addition, the rules explicitly and formally define the mapping behaviour. In systems where the data mapping is done by hand written sensor adapters this knowledge is implicitly coded into the adapter. Furthermore, the rules can be used to enrich the sen_sense model with additional information, like sensing time or the location of the sensor.

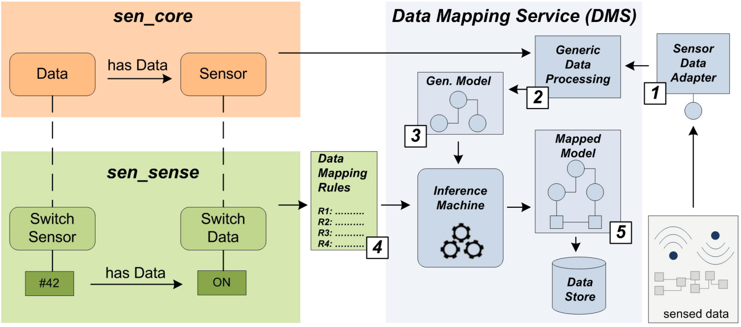


Fig. 1. Data Mapping Approach

As already mentioned we created two ontologies the sen_core ontology, which serves as basis for data mapping, and the sen_sense ontology, which covers the specific sensor environment. This was done for the following reasons: *(i)* the simple sen_core ontology serves as upper ontology and eases the integration of existing AAL systems [16], *(ii)* the sen_core ontology can be used as formal model to verify if the rules created a correct data model and *(iii)* the AAL specific ontology (sen_sense) can be changed without changing the basic infrastructure (sen_core ontology, data adapters, DMS)[2].

Figure 2 shows the SENIOR ontologies in more detail. We used the core ontology to define the basic objects *Sensor*, *Data*, *Time* and *Location* as well as basic relations among these objects. The concrete instances of the objects are defined in sen_sense. Each *Sensor* object is uniquely identified via an attribute (*hasIdAttr.*) and has a *hasData* relation to a *Data* object. On the Sen_core level the *Data* object only contains an attribute (*hasDataAttr.*) which stores

[1] In case that a new sensor which uses a currently not supported protocol is added, an adapter for this protocol must be created as well.

[2] Nowadays sensor networks offer sophisticated mechanisms for detecting faulty sensors. The rule based approach can additionally be adapted to avoid the corruption of the data store with incorrect or multiple (duplicated) sensed values by checking for faulty or duplicated sensor data.

the perceived sensor information as string. Via rules this unprocessed sensor data will be mapped to more specific data and sensor objects which are defined in the sen_sense ontology. Figure 2 shows this by using a *SwitchSensor* object for demonstration. The *Data* and *Sensor* objects are created by the DMS when an adapter perceives (pushes) new data. In case that complex calculations are necessary (data pre-processing) an adapter can in addition create a sen_sense based data model and push the model into the DMS. The two core ontology objects *Time* and *Location* are actually not used by the DMS. Instances of these objects are created by the mapping rules, which are currently used for: *(i)* mapping a sensor to its location, *(ii)* mapping sen_core *hasDataAttr.* values to sen_sense data objects (like the *ON/OFF* objects shown in figure 2) and *(iii)* creating temporal relations between data and sensor objects.

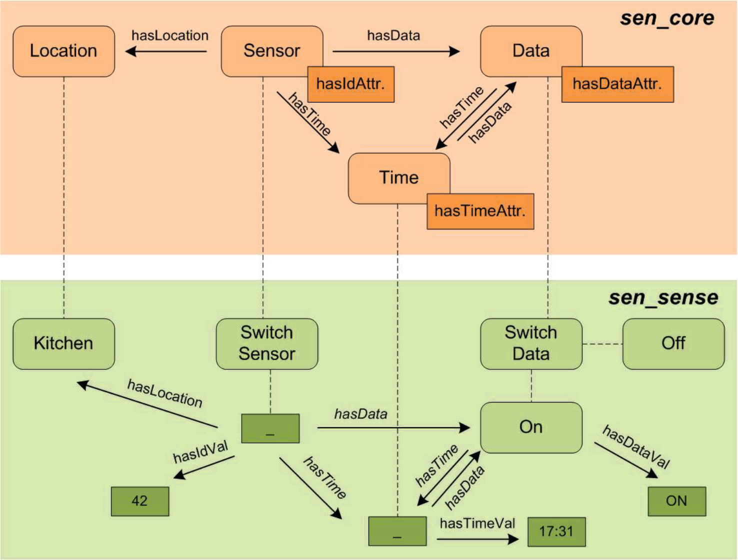


Fig. 2. Senior Core Ontology

Alternatively to our SENIOR Sense Ontology existing and sophisticated ontologies, like OntoSensor [12] could be reused. In case of OntoSensor, which is based on SUMO and SensorML, the *OntoSensor:_Sensor* class must be derived from *sen_core:Sensor* and the mapping rules need to be adapted correspondingly. Due to the amount of objects and relations defined in OntoSensor, the mapping rule files will become hardly comprehensible and understandable. Therefore, we decided to introduce a simple and manageable ontology for demonstration purposes which is directly based on sen_core.

5 Architecture and Implementation

We decided to use the service infrastructure provided by the OSGi framework to implement our prototype. Actually there is no common agreed standard for home automation systems, but a variety of projects, like the afore mentioned SOPRANO, OASIS and PERSONA systems are either based on OSGi or provide adapters for it. Besides, several commercial and open source implementations of the OSGi framework are available. As formalism for our ontologies (sen_core and sen_sense) the *Web Ontology Language (OWL)* [14] was used. For simple data mapping it would as well be possible to use the *Resource Description Framework (RDF)* [15]. Instead we decided do use OWL, because of the following reasons: *(i)* OWL is a widespread and machine processable notation for ontologies, *(ii)* there exist a variety of tools for editing and processing OWL, *(iii)* there are existing ontologies for Smart Home and AAL which can be re-used or integrated and *(iv)* OWL provides a description logic (OWL-DL) which can be used for reasoning.

As described in figure 1 we use an inference machine which creates the mapped data based on the automatically generated sen_core model and mapping rules. Thanks to the semantic web hype, there are several frameworks available which can be used for processing of and reasoning about OWL information. We decided to use the *Apache Jena Framework* [2] which provides a Java based API for OWL ontologies and a rule-based inference engine which offers sufficient reasoning support for our purposes.

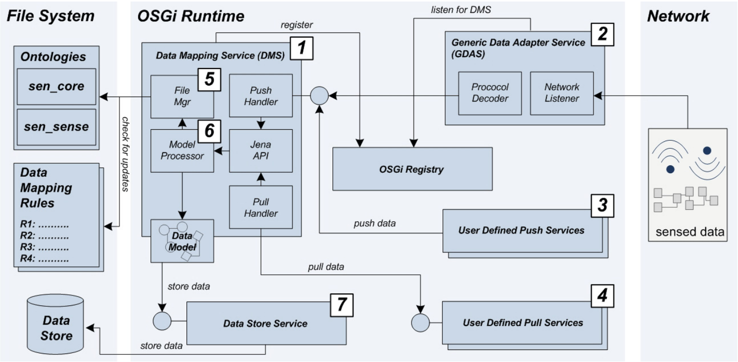


Fig. 3. Data Mapping Architecture using the OSGi and JENA Framework

Figure 3 gives an overview of the different service components used in our prototype. The *Data Mapping Service (DMS)* (1) was implemented as OSGi service and serves two purposes: *(i)* it provides an interface where either the *Generic Data Adapter Service (GDAS)* (2) or user defined adapter services (3) can push sensed data and *(ii)* it regularly polls registered pull adapter services (4) for data. The GDAS was introduced to avoid implementing a custom adapter for each sensor type. Actually it is used to process simulated sensor data which is generated by a simulator for ambient intelligence environments which was already presented in [6]. In addition, user defined adapter services can be provided for handling different sensor types and pushing data into the DMS. When data is pushed into the DMS a generic *Jena* model based on the sen_core ontology is created. Afterwards, the *FileManager* (5) checks, if suitable mapping rules are available for the given sensor input. The mapping rules are located in a designated, configurable folder on the file system which is regularly checked for updates. This allows to update the rules during runtime. The generic Jena model and the rule file are fed into the *ModelProcessor* (6). Jena provides the possibility to use different reasoners for inferring models, like Pellet, FaCT or the built-in GenericRuleReasoner. Depending on the rule description the created model can contain sensor data objects and additional information (for example, temporal or location information).

The mapped model is passed to the *DataStoreService* (7) which stores the Jena model into a database. In our prototype implementation we use the SDB API provided by Jena to store the data in the RDF format into a relational database. Currently different databases are supported by Jena. To ease the implementation we decided to use the simple *Hyper SQL* [13] for demonstration purposes.

Table 1. Rule snippet from a sensor mapping file

```
...
[Rule1:
  (?s sen_core:hasIdValue '701')
 ->
  (?s sen_core:hasLocation sen_sense:Kitchen)
]

[Rule2:
  (?s sen_core:hasIdValue '701')
  (?s sen_core:hasData ?d)
  (?d sen_core:hasDataValue ?rawData)
  (?currentTime rdf:type sen_core:Time)
  equal(?rawData, 'ON')
 ->
  (?s sen_core:hasData sen_sense:On)
  (sen_sense:On sen_core:hasTime ?currentTime)
  (?s sen_core:hasTime ?currentTime)
  (?currentTime sen_core:hasData sen_sense:On)
]
...
```

The mapping rules are currently stored in multiple files. For performance and flexibility reasons each sensor has its own rule file which defines the data and location mapping for the corresponding sensor. The rules are notated using the Jena Rules syntax. This offers the benefit that the file can be directly interpreted by the Jena API (no parsing needed). Figure 1 shows a simplified rule snippet from a mapping file. Rule1 is responsible for mapping the sensor with the id 701 to the location Kitchen. Rule2 is responsible for mapping the sensed raw data, which are stored as hasDataValue string property and can be interpreted as follows: If there is a sensor with id 701 and the property hasDataValue equals the string 'ON' four relations will be created. First a relation between the sensor and the object On. Second a relation between the On object and the current time. Third a relation between the sensor and the current time object. Forth a relation between the current time object an the On object. The mutual relations between data and time are of use on higher levels where, for example, property chains can be used to query if sensor-data-relations were valid during a defined time period. This example clearly shows that our mapping approach can also be used for enriching the sensor data model with additional information during the mapping process.

6 Evaluation and Conclusion

Target of this study was to develop a flexible approach for mapping sensor data to ontological models which is according to the *Data Fusion* community also known as *Object Refinement* (JDL data fusion model [10]). In our case sensed data is mapped to a formal model and enriched with a minimum of contextual information (sensing time and sensor location) before the data model is created. Situation Assessment, where relations among objects are evaluated with the goal to detect a defined situation, is not target of our work. This is done on superior levels of information processing.

To evaluate our approach we implemented a prototype based on the OSGi Framework. The *Data Mapping Service*, a storage service, a generic data adapter service, as well as a user defined sensor adapter service were implemented. The service-oriented implementation in combination with our mapping approach allows to add or remove sensors during runtime (starting or stopping adapter services for handling sensor input). We have tested this behavior via adding new types of sensor adapters during runtime. After the adapter bundle was installed in the OSGi target platform and a corresponding mapping file was found, the new type of sensor input was successfully processed by the system (i. e. the corresponding sen_sense sensor and data objects were created in our database). To test our implementation we used a simulator [6] containing a floor plan of a small flat which we equipped with different sensor devices (switch sensors, motion sensors, temperature sensors, etc.). A simulated interaction with 16 sensors was forwarded to our mapping service prototype where the raw sensor data was mapped to data objects, enriched which location and temporal information and stored in an RDF data store. Tests concerning the performance of the data mapping were executed and delivered feasible results. From data perception (in the DMS) to the availability of the data model the average mapping duration is between 15 and 22 ms, depending on the sensor type and complexity of the mapping rules. Table 2 contains an excerpt of the accumulated performance values per sensor type measured in the prototype environment.

Table 2. Mapping Performance

Sensor Type	*Sensor Id*	*No. of Signals*	*Processing Time (in ms)*		
			Min	*Avg*	*Max*
Motion Sensor	501	184	0	16	32
Temperature Sensor	503	162	0	15	63
Binary Switch	601	73	0	22	47
Contact Switch	604	64	0	22	47

Up to now the mapping rules only contain simple instructions. In case of a switch sensor this means that the data attribute *'0'* is mapped to the data object *ON* as well as introducing relations to location and time objects (compare figure 2). As future work it is planned to introduce and evaluate more complex mapping rules. Additionally, we want to increase the number of sensor instances and sensor types and evaluate its impact on the mapping performance. Concerning AAL systems which require an expandable mapping behaviour, we have demonstrated with our prototype that the presented approach is feasible and provides the required flexibility for dynamically expanding or changing the sensor components of an AAL system.

Acknowledgements. This research was funded by the European Regional Development Fund (ERDF) in cooperation with the Upper Austrian state government (REGIO 13). Any opinions, findings and conclusions or recommendations in this paper are those of the authors and do not necessarily represent the views of the research sponsors.

References

1. Amoretti, M., Wientapper, F., Furfari, F., Lenzi, S., Chessa, S.: Sensor Data Fusion for Activity Monitoring in Ambient Assisted Living Environments. In: Hailes, S., Sicari, S., Roussos, G. (eds.) S-CUBE 2009. LNICST, vol. 24, pp. 206–221. Springer, Heidelberg (2010)
2. Apache: Jena Framework (2012), http://incubator.apache.org/jena
3. Bauer, P., Rodner, T., Litz, L.: AAL-Eignung von Home Automation-Sensorik - Anforderungen und Realität. In: Proc. of Technik für ein selbstbestimmtes Leben - 5. Deutscher AAL-Kongress (2011)
4. Bekiaris, E., Bonfiglio, S.: The OASIS Concept. In: Stephanidis, C. (ed.) Universal Access in HCI, Part I, HCII 2009. LNCS, vol. 5614, pp. 202–209. Springer, Heidelberg (2009)
5. Buchmayr, M., Kurschl, W.: A Survey on Situation-Aware Ambient Intelligence Systems. Journal of Ambient Intelligence and Humanized Computing 2, 175–183 (2011)
6. Buchmayr, M., Kurschl, W., Küng, J.: A Simulator for Generating and Visualizing Sensor Data for Ambient Intelligence Environments. Procedia Computer Science 5, 90–97 (2011)
7. Chiriac, S., Saurer, B.: An AAL Monitoring System for Activity Recognition – Results from the First Evaluation Stages. In: Proc. of Technik für ein selbstbestimmtes Leben - 5. Deutscher AAL-Kongress (2011)
8. Esteban, J., Starr, A., Willetts, R., Hannah, P., Bryanston-Cross, P.: A review of data fusion models and architectures: towards engineering guidelines. Neural Comput. Appl. 14(4), 273–281 (2005)
9. Klein, M., Schmidt, A., Lauer, R.: Ontology-centred design of an ambient middleware for assisted living: The case of soprano. In: Proc. of the 30th Annual German Conference on Artificial Intelligence (2007)
10. Llinas, J., Bowman, C., Rogova, G., Steinberg, A.: Revisiting the jdl data fusion model ii. In: Proc. of the 7th Int. Conference on Information Fusion, Stockholm, Sweden, pp. 1218–1230 (2004)

11. OSGi Alliance: OSGi Framework (2012), http://www.osgi.org
12. Russomanno, D.J., Kothari, C.R., Thomas, O.A.: Building a sensor ontology: A practical approach leveraging iso and ogc models. In: Proc. of the Int. Conference on Artificial Intelligence (IC-AI), Las Vegas, US, pp. 637–643 (2005)
13. The hsql Development Group: HyperSQL (2012), http://www.w3.org/RDF
14. W3C: OWL 2 Web Ontology Language (2009), http://www.w3.org/TR/owl2-overview
15. W3C: Resource Description Framework (RDF) (2012), http://hsqldb.org/
16. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-based integration of information - a survey of existing approaches. In: Proc. of the IJCAI 2001 Workshop on Ontologies and Information Sharing, pp. 108–117 (2001)

Parameters for Service Level Agreements Generation in Cloud Computing

A Client-Centric Vision

Mariam Rady

Christian Doppler Laboratory for Client-Centric Cloud Computing,
Softwarepark Hagenberg, Austria
mrady@cdcc.faw.jku.at

Abstract. Current Service Level Agreements are lacking completeness. They often leave out parameters that are important for the consumer. In addition, SLAs are predefined by the service providers and the consumer does not have a say in them. The goal of this paper is to find the important parameters that should initially be included in a Service Level Agreement between a cloud service provider and a cloud service consumer. In this paper we are showing a client-centric view of the SLA. Some examples for SLAs of services in the market are illustrated, in order to show the need to specify, which parameters should be included in the SLAs.

1 Introduction

In this paper we aim at finding an initial set of parameters that should be present in a Service Level Agreement (SLA) to guarantee the Quality of Service (QoS) the cloud consumer is minimally expecting. We do that by reviewing the concept of non-functional requirements(NFRs) in Service Oriented Architectures(SOA) and applying it on cloud computing. We also present a short review on SLAs of existing cloud services, and how they lack various considerations. Most of the research on SLAs presents how to model SLAs assuming to have already defined the SLA parameters. Current SLAs as they are represented in the market lack preciseness and completeness. Some parameters that are important for consumers are left out and others are very vaguely described. In this paper we try to find out an initial set of SLA parameters necessary for cloud computing before modeling and operating on the SLAs.

In section 2.1 we will start by reviewing NFRs in literature, as we will use some of them as attributes to be included in SLAs. Section 2.2 is an overview on cloud computing and how SLAs are dealt with in this field. In addition some examples of currently existing SLAs in cloud computing are given. In section 3 we gathered a list of NFRs from different sources and tried to categorize them, and come up with a list of relevant NFRs to be included in the service contracts. These attributes are then analyzed for the cloud computing domain in section 4. In this paper we concentrate mainly on quality attributes that should be included

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 13–22, 2012.

in the SLAs in cloud computing. Nevertheless, we mention other categories that should be present in the contract such as legal issues and business values.

2 Literature Review

2.1 Non-Functional Requirements

For decades, researchers have tried to find a clear definition of NFRs. However, they still haven't reached consensus about what exactly NFRs are[11][14][17]. Questioning the users is never enough to determine the NFRs necessary to guarantee the quality the client is expecting[14][13][7].

Some of the work on NFRs, adapt the definition of parts of the quality model in the ISO/IEC 9126 [1] quality standards. The paper [5] uses the quality model as a basis for modeling of NFRs. This might be sufficient in the case of general specifications of NFRs, but with cloud computing advancing fast, new challenges arise e.g. Scalability [2], which is an attribute that is not part of the ISO/IEC 9126 quality model.

A taxonomy for identifying and specifying NFRs for SOAs was presented in [11]. Here NFRs are specified into three different categories: process requirements, non-functional external requirements and non-functional service requirements. In comparison to traditional software in SOA more NFRs occur due to the highly distributed nature of SOA[11]. For each of these categories a list of requirements with their definitions have been introduced[11]. Some of the attributes in this taxonomy will be used in our work to find appropriate SLA parameters to suit the client's needs in the cloud computing paradigm.

Relating NFRs to specific system domains can give insight how to define them. In [17] five types of systems were analyzed. The common NFRs between the different systems are: performance, reliability, usability, security, and maintainability[17].

A survey was made in [3] to create a list of NFRs from the consumer's point of view for SOA. The list consists of 17 NFRs. Some of these NFRs are actually of great importance to cloud computing.

Different quality attributes for SOA, the different factors related to each attribute and the existing efforts to achieve that quality were described in [18]. Most of the literature discussed NFRs for SOA. But with cloud computing emerging we need to analyze them for the cloud computing domain. These NFRs can then be used to express SLAs. The next section introduces cloud computing and how SLAs are represented in this domain.

2.2 Cloud Computing and Service Level Agreements

Cloud computing is the sum of Software-as-a-service (SaaS), Platform-as-a-Service (PaaS) and Public Utility Computing. It is providing computing, communication and storage at a particular price per hour. It provides the illusion of infinite computing resources available on demand, therefore resources are available when needed. It also allows the elimination of an up-front commitment by cloud users, so that companies can start small and increase their size once they are ready. In addition it provides the ability for paying "as you go", which means that cloud users are only paying for what they are using. Another advantage is that it removes the maintenance burden off the cloud consumer[2]. There are three types of cloud services that are usually discussed. These are Infrastructure as a Service(IaaS), Platform as a Service(PaaS) and Software as a Service(SaaS). A cloud consumer can be either a normal user or a developer that is aiming at developing a service on the cloud with the aim of being a service provider himself.

Generally, SLAs define assertions of a service provider, that the service he is providing meets a certain guaranteed IT-Level and business-process level service parameters. In addition, the service provider should provide measures to be taken in the case he failed to meet these assertions[16]. If we take a look at current SLAs for cloud services in the market they are all predefined and they mostly include the attribute availability in terms of monthly uptime percentage. However, there are key factors that are not included or very imprecisely described in the currently existing SLAs. We cannot deny that availability is one of the key qualities that need to be included in an SLA, but it's not the only one. For instance, how does the cloud provider guarantee performance, service maintenance and reliability? Is it enough to include in the customer agreements that the service provider would do everything he can to provide data privacy and confidentiality? All these questions lead us to think, that the current SLAs are definitely not representative enough of the clients' needs and what we need is a clear definition, what should be included in SLAs in cloud computing, so that consumers are satisfied. We present some examples on current SLAs in cloud computing.

Examples on Existing Service Level Agreements in the Cloud. We will take a look at IaaS, PaaS and SaaS. An example of IaaS is Amazon EC2. Examples of PaaS are Heroku and Google App Engine. SaaS are basically web services or web applications that we deal with on regular basis. In this section we do not only present SLA but also customer agreements and service terms, as it is often stated that the SLAs are governed by them.

Amazon EC2 is an IaaS service provided by Amazon Web Services. If we take a look at their SLAs, only availability is the issue there. They are promising a certain percentage of uptime. They are also describing the penalty Amazon has to pay in case of unavailability[20]. In the SLA they mention that the AWS Customer Agreement, Service Terms and Acceptable Use Policy are also included in the contract. If we take a look at the AWS Customer Agreements, they are mentioning the following attributes: security, privacy, confidentiality and some other legal issues[20], however, without measurable assurances or guarantees. The Acceptable Use Policy describes prohibited use of the service offerings[20]. AWS Service Terms of EC2 defines the responsibility of customers to maintain licenses and adhering to license terms of any software that is run on EC2[20].

Google App Engine -a PaaS- provides the customer with a Customer Agreement and an SLA. In the Customer Agreement they discuss primarily legal issues (e.g. Terms for Suspension and Removals, Intellectual Property Rights and Service Deprecation). In addition they mention terms for payment and penalties.

They also add terms for privacy and confidentiality. In the SLA they promise the customers a certain percentage of uptime and error rate calculation (part of Service Reliability)[8].

Heroku is also a PaaS. They do not provide SLAs but they call it "customer promise". They mention that the user is the one who is owning the code and the data on the platform and subsequently will have all the responsibility to protect it. In addition, they will try to offer 100% uptime. But if the service was down, they will provide the customer with an explanation.

3 Important Non-Functional Aspects for Cloud Computing- A Client Centric View

We started by looking at different lists of NFRs presented in[3][11][17][10][1][9]. We divided NFRs for cloud computing into 3 categories quality attributes, business values and legal issues, this is similar to the definitions in [11] and [10]. The first step we did is gathering the quality attributes for SOA to find all common attributes between the different taxonomies, definitions and classifications in our reviewed literature. We provide a list of different attributes that we claim are important from the consumer's point of view specifically for cloud computing. The paper [15] presents a framework to deal with quality attributes in software design and development. They present 7 different levels for handling the quality attributes in the different software development and design stages. The first level is the specification of the quality properties, this includes a semi-formal definition, a measurement and policies for evaluation. However, our aim here is not to define the quality attributes individually, but rather focus on the bigger picture, which of these attributes is useful in the cloud computing domain. This is why, we do not include the measurement and the evaluation policies, although they are stated in the papers [18][3][11] discussing the different non-functional aspects. The different quality attributes are presented below.

Usability is the measure of the quality of user experience in interacting with the service[18]. It is the ability of the service to be understandable, learnable, operable and attractive to a consumer[3][1]. Usability is needed whenever users are dealing with interfaces.

Reliability is the ability of a system to keep operating over time without failure. There are two types of reliability [18]:

Message Reliability: services typically communicate with each other or with consumers through messages. These are dependent on the network performance. This means that if the connection channel is not reliable, then message delivery assurance is necessary.

Service Reliability: the service operates correctly with transactions preserving data integrity and if it fails it reports failure to the user [18].

Performance includes the following attributes

Response time: the duration of time between sending a request to a service and receiving a response[10].

- **Throughput:** the amount of requests a service can handle in a certain time[10].

Security includes the following attributes

- **Authenticity:** trusting that the indicated identity of a subject is true[18].
- **Integrity:** impossible change or deletion of data by unauthorized subjects[11].
- **Confidentiality:** access to data is given only to authorized subjects[18]. Unauthorized individuals are denied access to data [3].
- **Privacy:** the ability of a subject to control sharing personal information[19].
- **Auditing:** performing logging and assuring non-repudiation of subjects' actions [10][3].

Interoperability is the ability of communication entities to share specific information and operate on it according to an agreed-upon operational semantics [6]. In order to achieve interoperability standardization is needed. In addition, in SOA as well as cloud computing users deal with service interfaces that provide an abstraction of the process that is going on behind the scenes[4], which can help in achieving interoperability.

Availability defines the amount of time the system is operational and accessible when required for use. In cases of downtime service providers generally pay penalties in different forms for consumers[18].

Some attributes were not presented in all the NFR-lists in our reviewed literature. However, we think that they are important for cloud computing. These attributes are:

Scalability is the ability of services to function properly when the system needs to be changed in size. Scalability can follow through horizontal scaling, vertical scaling or stateless services[18].

Portability As cloud computing services are accessed over the internet through interfaces, service consumers need to be sure that the services will be working on different devices or on different platforms.

The next three attributes are more a concern of service providers, however they indirectly affect the quality of the cloud product, as they affect its agility. This facilitates maintenance and updating of the cloud product.

Modifiability is the ability to make changes to the service efficiently and with low costs [11][9].

Maintainability is the ability to maintain the service[3].

Testability is the ability to test the service is working according to the service requirements[11].

For **Business Values** we gathered the following attributes:

Price in cloud computing is a unit price per hour for usage of the service. The ”Pay as you go” concept[3].

Penalty and Compensation Penalty is the fee a consumer has to pay when he breaks the contract with the service provider. And compensation is the fee the service provider has to pay if he breaks the contracts with the consumer[10].

And important **Legal Issues** are:

Jurisdiction Systems need to comply with legislation of the country/territory they are hosted in. Services should provide their locations to reflect the legal obligations the consumer would have if he used the service[3].

Termination and Suspension are the terms for terminating and suspending the service[8].

Note that in the literature availability is sometimes considered as a subquality of security[18]. But here we chose to deal with it seperately as availability is one of the main qualities that need to be defined for cloud computing. As in the different SLAs one of the main attributes is availability.

In the next section we present how we used the previously listed NFRs to find the important parameters that should be included in SLAs of cloud computing services. In addition, we give reasons why each of the previously listed attributes is important for cloud computing.

4 Important Parameters for Service Level Agreements in Cloud Computing

In cloud computing quality attributes are depending on the type of service it is providing. Cloud services can provide different layers of the IT-Stack as a service. The responsibilities and concerns of a service provider as well as the consumer differ according to which layer of the IT-Stack the cloud computing service is offering. Not only in the amount of attributes but also how they are being dealt with. In this section we are trying to figure out the responsibilities of the cloud provider and the cloud consumer for different services. As we are investigating the view point of the cloud consumer, we do not consider the fact that a service provider is himself a user of another cloud service that is supporting the service he is providing. The guarantees the cloud provider is giving should be stated upfront to the cloud consumer, regardless of the fact that a third party is involved in the service he is providing. Figure 1 shows the IT-Stack and how the different cloud services will be concerned with different management issues accordingly. In Infrastructure as a Service (IaaS) the consumer has greatest responsibilities for management, from the application layer all the way down to the top of the virtual infrastructure. Entering a cloud at the level of IaaS is replacing having datacenters on premises. Instead, the consumer is provided with virtual machines, virtual storage, networking. It is then the job of the consumer to build everything on top of it, to select the operating system, the middleware and pay for it if necessary. Then the consumer is able to use and develop his own solutions[12].

When entering the cloud at the SaaS level, the consumer is basically selecting a cloud application. Some of these cloud applications in SaaS can be customized according to the costumers' needs[12].

Platform as a Service is in the middle. The consumers' responsibility is the whole application layer down until the top level of the middleware. It is a platform on which the consumer can build his own application and create his own

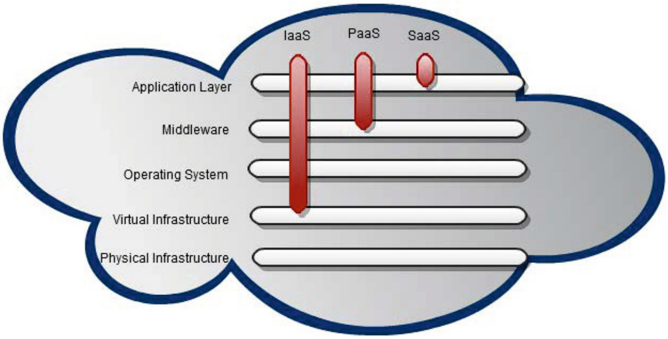


Fig. 1. Cloud Computing Services in the IT Stack

unique solution. It is providing a middleware with its elasticity, scalability and availability. These are attributes that the consumer doesn't have to care about because they are already built on PaaS[12].

As the focus of the different cloud services differ we claim that the SLA parameters would also be different among the cloud services. Because as we go from the IaaS to SaaS the responsibilities are shifting from the cloud consumer to the cloud service provider.

If cloud providers want to encourage companies to move to the cloud they have to guarantee at least the same quality of service they are getting on premises. After listing the different NFRs with their QoS attributes for cloud computing we will analyse them for the different services in cloud computing. And we will list the different parameters that should be initially included in the SLAs.

If the cloud service is accessible through an interface, then the cloud service provider should guarantee the following attributes:

Availability: The interface should be available all the time.

Usability: The interface should be user-friendly, learnable and understandable, in the case the service is operated through a graphical user interface.

Reliability: The interface should provide reliable customer-interface communication, and interface-cloud communication. It should also provide error messages when failure happens.

Security: Unauthorized access to the service should not be allowed. By that the service provider can guarantee confidentiality of the consumer's data.

Interoperability : Allowing the service interface to interact with other services.

For the cloud service the obligations and the responsibilities of providers and consumers differ based on the type of the cloud service they are offering or using respectively. And thus the quality requirements for each of the cloud services also differ.

In IaaS the service provider provides various resources. He should guarantee *performance* of the hardware and virtual infrastructure just like it would be operating on premises. The *reliability* of hardware and virtual infrastructure is also an important quality, the hardware and virtual infrastructure should operate exactly as expected. The hardware and virtual infrastructure should be available all the time and the *maintenance* of the hardware and virtual infrastructure is the responsibility of the service provider. It is assumed that the cloud consumer can have horizontal *scalability.* However, if he needs vertical *scalability*, this has to be agreed on in the contract.

The consumer has the responsibility of whatever is deployed on that infrastructure. But for that he has the freedom to choose his operating system, his middleware and the applications he would want to run on top of this infrastructure, just as he would do it locally.

With PaaS the service provider has more responsibilities for the service and the consumer has less responsibilities than in IaaS. He also needs to guarantee the four attributes that the service provider of IaaS is guaranteeing in addition to other attributes but the context in which these attributes are guaranteed is completely different. The service provider should guarantee the *performance* of his service, not only the performance of the platform but also indirectly the performance of the underlying hardware and virtual infrastructure infrastructure. The *reliability* of the platform and the underlying infrastructure should also be guaranteed. In addition the service provider should guarantee *availability* of the platform and its underlying hardware and virtual infrastructure. *Maintenance* of the platform and the underlying infrastructure is also the service provider's responsibility.

Added to that the service provider should guarantee *scalability* of the underlying structure so that the consumer can manage his application as he will not be gaining access to the virtual infrastructure as it is the case for the IaaS. If the consumer decided to develop his solutions on the cloud then he should get the illusion of infinite computing as mentioned in section 2.2. And the platform should be able to deal with the change in size.

Moving to SaaS here the service provider has the most responsibility. All the quality attributes that are mentioned in section 3 are here the service provider's responsibility. The software is no longer installed locally by the client but the client can access it on the cloud through a service interface. The software that is provided as a service has to provide good *usability,* it has to be easy to learn, to understand and to operate. The service has to provide *reliability*, the customer needs to be sure that the software he is signing up for will operate correctly. The consumer also needs to know that the service will provide good *performance*, and that he will not need to compromise in terms of response time and throughput when moving from a locally installed software to the cloud. The *security* of the data is here the responsibility of the service provider. In addition, *availability* is one of the key qualities that need to be ensured for SaaS. The consumer should be able to rely on the fact that the service will be available all the time. In addition, the consumer needs *maintenance* for the service. *Interoperability* allows communication of services and is one of the advantages of using a SOA, which is essentially what SaaS is. The consumer will also need a *portability* guarantee, that the service can work in different environments. There are some attributes that were mentioned in section 3 that are the concern of the service provider but are no longer the consumer's concern in SaaS. However, they indirectly affect the consumer's concerns. These are scalability, modifiability, testability and maintainability. These do not need to be included in the SLA but we would like to draw the attention of service providers to them, because these will improve the agility of the service e.g. updatebility.

For each of the previously mentioned cloud services the service provider should include the relevant quality attributes in the SLAs and provide the measures that he is going to take to guarantee the quality of service.

Business values and legal issues will be included equally for IaaS, PaaS and SaaS. These however, are not the current focus of this paper.

5 Conclusion and Future Work

This paper explains how currently existing SLAs do not fully represent the consumers' needs, as they are usually predefined by the service providers and the consumers do not contribute in generating them. We started our work by taking a look at the different NFRs and finding the needed NFRs for cloud computing. After having a list of different NFRs we needed to figure out which of these should be included as SLA parameters. For different cloud services different SLA parameters should be included. We presented the three most discussed service types in cloud computing, namely IaaS, PaaS and SaaS. Starting with IaaS that provides a virtual infrastructure to the consumer and here we've seen that the responsibilities of the service providers are limited to the hardware and virtual infrastructure they are providing. In PaaS the service provider is providing more of the IT-Stack and this is why he also should provide more guarantees regarding the quality of the services he is providing. Coming to SaaS, the service provider is basically providing the whole IT-Stack and in that case even more qualities are expected from him. For our future work we plan on formally defining the SLA parameters individually for each of the cloud services, define the different metrics for the different parameters and measures that need to be taken in order to guarantee the quality of service that the cloud user is expecting. In order to encourage consumers, especially enterprises to go to the cloud, the cloud providers should at least guarantee the same quality of service that the enterprise is having on premises.

References

1. ISO/IEC 9126. Software engineering, Product quality -Part 1: Quality model
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A berkeley view of cloud computing. Technical report (2009)
3. Becha, H., Amyot, D.: Non-Functional Properties in Service Oriented Architecture A Consumer's Perspective (2011)

4. Bergholtz, M., Andersson, B., Johannesson, P.: Abstraction, Restriction, and Co-creation: Three Perspectives on Services. In: Trujillo, J., Dobbie, G., Kangassalo, H., Hartmann, S., Kirchberg, M., Rossi, M., Reinhartz-Berger, I., Zimányi, E., Frasincar, F. (eds.) ER 2010. LNCS, vol. 6413, pp. 107–116. Springer, Heidelberg (2010)
5. Botella, P., Burgues, X., Franch, X., Huerta, M., Salazar, G.: Modelling Non-Functional Requirements. In: Proc. of Jornadas Ingeniera de Requisitos Aplicados, JIRA (2001)
6. Brownsword, L., Carney, D., Fisher, D., Lewis, G., Meyers, C., Morris, E.J., Place, R.H.: Current perspectives on interoperability. Technical report, Carnegie Mellon Software Engineering Institute (2004)
7. Chung, L., Leite, J.C.S.P.: On Non-functional requirements in the Software Engineering. Software Quality Journal 5(4), 285–294 (2009)
8. Google App Engine. Service Level Agreement
9. Software Engineering and IEEE Standards Committee. IEEE Recommended Practice for Software Requirements Specifications 1993 (1993)
10. Advancing Open Standards for the Information Society (OASIS). Web Services Quality Factors Version 1.0, pp. 1–29, (July 2011)
11. Galster, M., Bucherer, E.: A Taxonomy for Identifying and Specifying Non-Functional Requirements in Service-Oriented Development. In: 2008 IEEE Congress on Services - Part I, pp. 345–352 (July 2008)
12. Gartner and IBM. Explore PaaS with a video from Gartner and IBM
13. Glinz, M.: On Non-Functional Requirements. In: 15th IEEE RE Conf. (2007)
14. Grimshaw, D.J., Draper, G.W.: Non-functional requirements analysis: deficiencies in structured methods. Information and Software Technology 43(11), 629–634 (2001)
15. Jaakkola, Thalheim: Framework for high-quality software design and development: a systematic approach. IET Software 42, 105–118 (2010)
16. Ludwig, H., Keller, A., Dan, A., King, R.P., Franck, R.: Web Service Level Agreement (WSLA) Language Specification. pp. 1–110 (2003)
17. Mairiza, D., Zowghi, D., Nurmuliani, N.: An Investigation into the Notion of Non-Functional Requirements. In: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC 2010, p. 311 (2010)
18. O'Brien, L., Merson, P., Bass, L.: Quality Attributes for Service-Oriented Architectures. In: International Workshop on Systems Development in SOA Environments (SDSOA 2007: ICSE Workshops 2007), p. 3 (May 2007)
19. University of California. Privacy and Confidentiality (retrieved: March 2012)
20. Amazon Web Services. AWS Customer Agreement and Service Level Agreement

Resource, Process, and Use – Views on Service Modeling

Birger Andersson, Maria Bergholtz, and Paul Johannesson

Stockholm University,
Department of Computer and Systems Sciences, Sweden
{pajo,ba,maria}@dsv.su.se

Abstract. Currently there exists a multitude of views of service creating problems for designers and users of models with respect to reasoning, description and classification of services. The diversity of conflicting views and definitions suggest that a multi-perspective approach is required to explicate the notion of service. The purpose of this paper is to introduce an integrated view of the service notion based on a literature survey. Our approach is to start with an analysis of service-as-a-resource and argue that this view will benefit from being complemented with a service-as-a-process view. These views are then integrated and represented in a conceptual model.

1 Bakground

The increasing interest in services has created a multitude of alternative views and definitions, often conflicting, of the service concept. What constitutes a service is still a matter of debate, in industry as well as in various research communities. The lack of a common view of the service concept makes it difficult to reason about, describe and classify services in a uniform way. One approach to structuring services is to divide them into business services and software services, and think about those much in the same way as any tradable resource.

An alternative to identifying services by their internal properties that uniquely distinguish them from other kinds of resources is to view services as perspectives on the use and offering of resources [Edv05]. Thus, the focus is shifted from the internal characteristics of resources to their context of use and exchange. This view is shared by the Unified Services Theory (UST) [SF06], which also bases its definition of services on the use and exchange of resources. UST defines service processes as processes where customers provide significant input resources, as opposed to non-service processes where customers only select what output resources to buy and pay for.

The diversity of service views and definitions, and the fact that these views are often conflicting, suggest that a multi perspective approach is required to explicate the notion of service. The purpose of this paper is to introduce an integrated view of the service notion based on a literature survey. Our approach is to start with an analysis of service-as-a-resource and argue that this view will benefit from being complemented with a service-as-a-process view. These views are then integrated and represented in a conceptual model. Furthermore, to capture use we introduce rights into the model. We claim that such an integrated view facilitates the understanding of models that include

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 23–33, 2012.

services, for designers as well as users. The work reported here is mainly a consolidation and integration of previous works [BAJ10], [BAJ11] and [WJBA09].

This paper is structured as follows. Section 2 briefly outlines the main points of the REA business ontology [Mc82] and Hohfeld's [Hoh78] classification of rights. In Section 3 we suggest a service classification, which in Section 4 is consolidated into a unified service model. Section 5 concludes the paper.

2 Resource, Process, and Right

In the following, we make use of the language provided by the REA business ontology when discussing (service) resources. The REA (Resource-Event-Agent) ontology was originally formulated in [Mc82] and developed further in a series of texts, e.g. [Hr06]. The ontology is based on the core concepts of resources, events, and agents. Originating in accounting, it also carries the underlying principle of duality in resource transfers between agents. An agent gives a resource expecting to get a resource in return. For service process we additionaly use SOA, service-dominant logic, and some of their precursors [WS04, OA06, Pr04, Lusch08, UN08, Zei85]. Hohfeld's classification of rights [Hoh78] is used as a means for analysing what kinds of rights are transferred in exchanges of services and other kinds of resources. For collaborating agents the distribution of rights to resources and processes is instrumental for their use.

2.1 Resource

A *resource* is something that is of value for at least one agent, e.g., a car or Internet access. Based on the degree to which a resource is tied to an agent, resources can be classified in three ways: independent resources, internal resources, and shared resources.

An *independent resource* is a resource that can exist independently of any agent. Typical examples of independent resources are physical objects, land, and information.

An *internal resource* is a resource that is existence dependent on one single agent. If the agent ceases to exist, so does the internal resource. Examples of internal resources are capabilities, skills, knowledge, and experiences. A characteristic of an internal resource is that is not an economic resource, i.e. it is non tradable.

A *shared resource* is a resource that is existence dependent on two or more agents. Common shared resources are relationships and rights. Some relationships are narrow in scope, primarily governing and regulating activities for some particular resource(s), e.g. ownership of goods or a purchase order. Other relationships have a wider scope, e.g. a marriage or an employment relationship that includes a number of rights. Rights are further discussed in Section 2.3.

In contrast to the above non-abstract resource classification an *abstract resource* is a resource that is not defined by its distinguishing properties or components but only through its use, i.e. what benefits it can bring to other resources or agents when it is used.

Finally, an *economic resource* is a resource on which rights can be transferred from one agent to another.

2.2 Process

Resources are not unchanging but can be transformed, i.e. they can be produced, modified, used, or consumed as well as exchanged between agents. This is done in conversion processes or exchange processes. In REA a *process* is defined as a set of *events*.

2.2.1 Conversion Process

Resources are transformed in so called conversion processes consisting of conversion events. A *conversion event* represents a transformation of a single resource. If the conversion event creates a new resource or increases the value of an existing resource, we say that the conversion event is a *production event*. If the conversion event consumes a resource or decreases the value of a resource without consuming it, we say that the conversion event is a *consumption event* or a *usage event*, respectively. Usage events are using resources that may be reused in several conversion events, (similar to the concept of 'asset' [Fo97]), while consumption events use up resources (similar to the concept of 'consumable' [Fo97]). Examples of conversion events are the production of bread, the repair of a car, and the consumption of a liter of fuel.

A *conversion process* is a set of conversion events including at least one production event and at least one consumption or usage event. The latter requirement expresses a duality relationship between production and consumption/usage events, stating that in order to produce or improve some resource, other resources have to be used or consumed in the process. For example, in order to produce a car, a number of other resources have to be used, such as steel, knowledge, and labour.

2.2.2 Exchange Processes

Resources can also be exchanged in exchange processes that occur between agents. An *exchange event* is the transfer of rights on some resource to or from an agent. If the exchange event means that the agent receives rights on a resource, we call the event a *take event*. If the exchange event means that the agent gives up rights on a resource, we call the event a *give event*.

An *exchange process* is a set of exchange events including at least one give event and one take event. Similarly to conversion processes, this requirement expresses a duality relationship between take and give events – in order to receive a resource, an agent has to give up some other resource. For example, in a purchase (an exchange process) a buying agent has to provide money to receive some goods. Two exchange events take place in this process: one where the amount of money is decreased (a give event) and another where the amount of goods is increased (a take event).

2.3 Rights

As a more precise understanding of rights will be required for characterizing different kinds of resources and exchanges, we here introduce a rights classification based on the work of W. N. Hohfeld, [Hoh78]. He identified four broad classes of rights: claims, privileges, powers, and immunities (immunity is not used in this paper).

One agent has a *claim* on another agent if the second agent is required to act in a certain way for the benefit of the first agent. Conversely, the second agent is said to have a duty to the first agent.

An agent has a *privilege* on an action if she is free to carry out that action without any interference from the environment in which the action is to be carried out. By environments is here meant social structures such as states, organizations or even families.

A *power*, finally, is the ability of an agent to create or modify a relationship.

3 Service Classifications

Based on the above and surveying literature on services [UM03, Pr04, WS04, VL06, OA06, Sp08, WJAB09, among others] it is possible to identify some salient characteristics of services, in particular:

(C1) A service is an economic resource, since it is an object that is considered valuable by actors and that can be transferred from one actor to another;
(C2) A service is always provided by one actor for the benefit of another actor;
(C3) A service encapsulates a set of resources owned by the provider. When an actor uses a service in a process, she actually uses the resources encapsulated by the service without getting ownership of these;
(C4) Service providers and customers co-create value together as both of them provide resources to be used and consumed in service processes;
(C5) A service is existence dependent on the processes in which it is produced and consumed, which means that the service exists only when it is consumed and produced. It is consumed and produced simultaneously. In contrast to goods and information, a service cannot be stored for later consumption.

According to C3, services can be used for restricting access to resources. An agent can provide access to her resources to another agent for instance by transferring the ownership of the resources of by using a service. Using a service instead of another kind of resource provides several benefits, as the service customer will not own the service. This means that she does not need to take on typical ownership responsibilities, like infrastructure management, integration, and maintenance. Instead, she can focus on how to make use of the service for satisfying her needs. For example, a person can satisfy her transportation needs either by buying and driving a car or by using a taxi service. In the former case, she will own the car required for the transportation, meaning that she will be responsible for cleaning it, repairing it, getting the right insurances, and many other infrastructure and maintenance tasks.

When using a taxi service, on the other hand, she does not have to care about any of those responsibilities but can focus on how to use the taxi to best satisfy her transportation needs. Thus, services provide a convenient way of offering and accessing resources by allowing agents to use them without owning them.

Services can provide an abstraction mechanism, according to C3 where resources are specified through their function and not their construction. In other words, a resource is defined in terms of the effects it has in a process, not in terms of its properties or constituents. To be able to offer resources in an abstract way provides several advantages. It becomes easier for a provider to describe the benefits of an offering when she can focus on the effects of the resource offered and abstract away from its accidental features. The provider can address the needs and wants of the customer and clarify how these are fulfilled by her offering without going into detail about its composition. Furthermore, the provider does not have to commit to any specific way of delivering her offering; instead, she can choose to allocate the resources needed in a flexible and dynamic way.

C4 takes its starting point in the observation that for most kinds of goods, customers are not involved in their production. The only role of the customer is to select which goods to purchase and pay for them. Instead goods are produced internally at a supplier who later on sells the goods to a customer who uses them without the involvement of the supplier. In contrast to a goods producing company, a service provider always has to work closely with its customers. In fact, a service can never be carried out by a provider in isolation as it always requires a customer to take part in the process. In such a service process the provider and the customer together co-create value as both of them provide resources to be used or consumed in the process. For example, in a photo sharing service the service provider will supply hardware and software, while the customer will provide photos and labour. Together they engage in a process that results in value for the customer – shareable photo albums.

In order to make the concept of service as co-creation more precise it is useful to distinguish between service as a process and service as a resource. The word “service” is sometimes used to denote a process, e.g., in the phrase “Today, our company carried out 25 car repair services”. In other cases, “service” is used to denote a resource, e.g., “Our company offers car repair services for the fixed price of 200 euros”.

The conception that a service is an economic resource implies that services can be exchanged by agents. A starting point for a classification is the recognition of what is called Core services [WJAB09]. Core services for an agent in a network specify what economic resources the agent is prepared to exchange with other members of the network.

Given a set of core services there are a number of services that add to or improve on these. These additional services are divided into four classes: complementary services, enhancing services, support services, and coordination services.

Complementary service. A service complements another service if they are part of the same service bundle and concern the same resource [We07]. For example, a gift-wrapping service complements a book sales service by having as goal to improve the book by packaging it in an attractive way. Thus, both services concern the same resource – the book. A service bundle is defined as the services provided to a customer in the same exchange process.

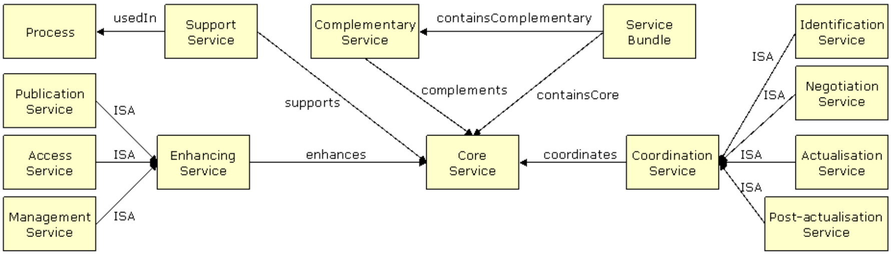


Fig. 1. Service Classification

Enhancing service. An enhancing service is a service that adds value to another service (rather than to some other kind of resource). The possibility of enhancing services follows from a conceptualization of a service as a resource. The enhancing service has an effect on the quality of another service or some feature like visibility or accessibility. By definition, it is existence-dependent on the other service. The following types of enhancing services can be identified:

- **Publication service.** A publication service provides information about another service (or any other resource) e.g. by means of a web page, a TV ad or a public service registry. Hence, it produces visibility of the service. At the same time, it increases the knowledge of customers, so it has a dual focus.
- **Access service.** An access service gives an agent access to another service, i.e. the agent uses the access service to invoke the other service. An advantage of using an access service is that it can act like a Façade object in Software Engineering [Ga95] that induces loose coupling by hiding the service details from the consumer. At the same time, it can contain medium-specific logic.
- **Management service.** A management service is a service that aims at maintaining or optimizing another service.

Support service. A service A supports a service B if A has as goal to produce B, or if A has as goal to produce a resource that is used in a process that produces B [Er07].

Coordination service. A coordination service is any service that supports (used in) an exchange process. It is used for ensuring that communicating parties in a business relation are coordinated or synchronized. The resource exchanged can be a service but also a good. Coordination services can be classified according to the stage in a business relation where the stages are identification, negotiation, actualisation, and post-actualisation [UM03]. For example, a catalogue service is instrumental in the identification stage. In the negotiation stage the terms and conditions of resource deliveries are formed (negotiation service, brokerage service) or reservations are made (reservation service). The actualisation stage is concerned with the actual deliveries of offered resources, including payment (payment service) whereas the post-actualisation stage may include all kinds of in-warranty services.

4 A Unified Service Model

In this section, we will introduce a conceptual model for services that integrates the resource perspective of a service with the process perspective. The model takes as point of departure the classification of services from Figure 1 [WJAB09] and complements it with service perspectives based on the ways resources can be used and exchanged in [BAJ11]. Hence the model does not include the term “service”, but instead a family of related terms, including “service resource”, “service offering”, and “capability”.

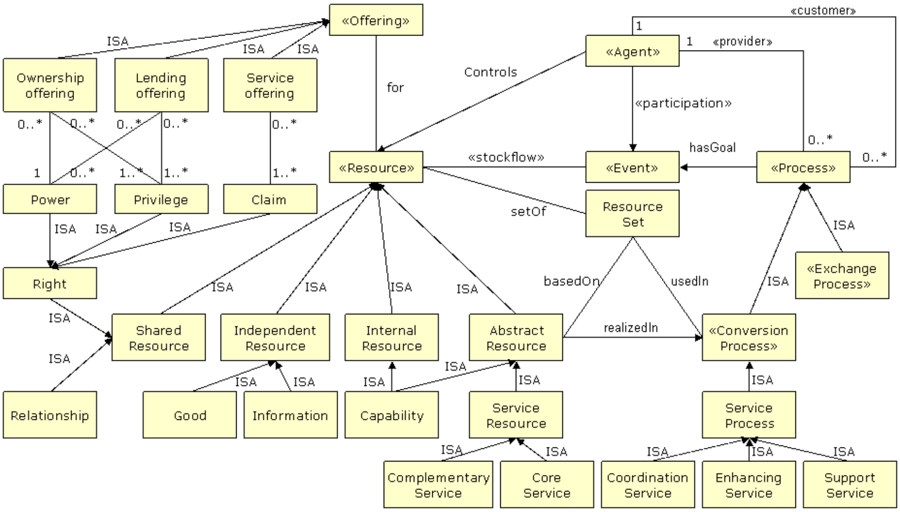


Fig. 2. Basic Service Ontology (core REA concepts in guillemots)

4.1 Service Resource

A service resource is an abstract resource that is defined only through its use and effects in a service process, i.e. what changes it can bring to other resources when consumed in such a process. For example, a hair cut service is defined through the effects it has on the hair style of a person. It is not defined by means of the concrete resources used when cutting the hair. The concrete resources to be used are left unspecified and can change over time. On one day the hair dresser may use scissors and shampoo and on another day an electric machine and soap, but in both cases he provides a hair cut service. Thus, the same service resource can be based on different sets of other resources, as shown in Figure 2, and when it is consumed exactly one of these resource sets will be used.

Service resource is modeled as a subtype of Abstract resource. As such, it automatically inherits all features of resources, in particular it can be offered by an agent, it is realized by a (conversion) process, or it can be used within a (conversion or exchange) process.

Services are exchanged like other resources in an exchange process that meets the REA duality principle. This exchange process needs to be distinguished conceptually from the process realizing the service, but they are interwoven in time. In the bottom of Figure 2 we find resources such as capabilities and service resources that are used or consumed in a service process (a conversion process) for the benefit of the customer that is involved in an exchange of value (usually money in return for a service resource) with the service provider.

While the notion of service resources primarily is useful for providing interfaces between agents in the context of resource exchanges, the related notion of capability can help to structure an organization internally. A capability is an internal resource that is defined through the conversion processes in which it can be used. Similarly to a service resource, a capability is abstract in that it is defined only by its use and the effects it can produce. In contrast to a service resource, a capability can be used in any process, not only in a service process. Thus, a capability of an agent can be used to produce something that is under the control of that agent. Furthermore, a capability is not an economic resource, i.e., it cannot be traded. Capabilities are often broadly and vaguely delimited, thereby specifying in general terms what an agent is able to accomplish. Service resources, on the other hand, are typically more precisely delimited as they are to be traded. Therefore, service resources are often used to externalise capabilities by exposing some parts of them.

Figure 2 shows three different ways for an Agent to make her resources available to other agents through offerings: an agent may offer to sell a resource to another agent, i.e. to transfer the ownership of the resource to the other agent, as modelled by Ownership offering. A transfer of ownership means that a number of rights are transferred from seller to buyer, in Figure 2 modelled by the class Right. The rights transferred include powers and privileges according to Hohfelds's classification (section 2.3). As an example, an agent offering to sell a book to a customer means that the agent is offering the customer privileges to use the book as well as the power to transfer the ownership of the book to yet another agent if she so wishes.

An agent may make an offer to lend a resource or provide access to it in a Lending offering. This means to offer an agent to get certain privileges on the resource for a period of time but without getting any ownership, i.e. the borrower is not granted the power to change the ownership of the resource. Optionally, the borrower may get some other powers, such as lending the resource to a third agent.

An agent may make a Service offering to a potential customer, which is the most abstract way of providing access to an agent's resources. A service offering means that the provider offers to use some of her service resources in a service process that will benefit the customer. Effectively the provider restricts access to these resources. In particular, the customer is not offered any powers or privileges on any concrete resources.

Core service and Complementary service from Figure 1 are Service resources, e.g. abstract recourses that are defined through the value they can provide to other resources or agents when they are consumed in service processes.

4.2 Service Process

A service process is a conversion process that uses or consumes resources from two agents, called provider and customer, and produces resources that are under the control of the customer, i.e. the customer gets rights on these resources, cf. C4 and C5. The provider in the service process has to actively participate in the process, while the customer may be passive. For example, a customer driving a borrowed car is not a service process, while a customer being driven by (a representative of) the provider is. Thus, a service process differs from other processes in three ways. First, some of the input resources are under the control of one agent, the provider, while the output resources are under the control of another agent, the customer. This means that the provider uses or consumes her resources in the service process for the benefit of another agent. Secondly, not only the provider but also the customer provides resources as input to the service process. Thirdly, the provider actively takes part in the service process.

A distinctive feature of a service process is that it has a goal to modify (and hence add value to) other resources. This is modeled by the association hasGoal from Process to Event. For example, the goal of the hairdressing service is to convert the customer’s hair. A service does not specify how it is to be realized, i.e. how its goals are to be achieved. Instead, a service can be realized in many different ways. To realize a service, the process must achieve at least the goal of the service. To be precise, the event being the goal of the service is contained in the process realizing it.

Coordination service, Enhancing service, and Support service from Figure 1 are all service processes. Their executions do in various ways coordinate, enhance, or support a conversion process that realizes an Abstract resource.

5 Concluding Discussion

In this paper we have presented an integrated conceptual model of services. This model integrates two common views on services namely service as a resource and service as a process. The service as a resource emanates from the goods-dominant logic whereas the service as a process emanates from service dominant logic and the differences in scope in these two formalisms make it cumbersome to reason about particular characteristics of services. For instance, using goods-dominant logic make it problematic to reason about co-creation of value, the latter which is a characteristic of a service dominant logic. Related to this point is the more general question on how to handle composition of services described from different viewpoints. This is in part a methodological question which we need to defer for future research. We do however believe that there is need for integration and that integrating the two views contributes to a basis for a better conceptual modeling of services.

References

[Ag08] Arsanjani, A., et al.: SOMA: A method for developing service-oriented solutions. IBM Systems Journal 47(3), 377–396 (2008)

[BAJ10] Bergholtz, M., Andersson, B., Johannesson, P.: Abstraction, Restriction, and Co-creation: Three Perspectives on Services. In: Trujillo, J., Dobbie, G., Kangassalo, H., Hartmann, S., Kirchberg, M., Rossi, M., Reinhartz-Berger, I., Zimányi, E., Frasincar, F. (eds.) ER 2010. LNCS, vol. 6413, pp. 107–116. Springer, Heidelberg (2010)

[BAJ11] Bergholtz, M., Johannesson, P., Andersson, B.: Towards a Model of Services Based on Co-creation, Abstraction and Restriction. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 476–485. Springer, Heidelberg (2011)

[Er07] Erl, T.: SOA: principles of service design Prentice-Hall (2007)

[Edv05] Edvardsson, B., Gustafsson, A., Roos, I.: Service portraits in service research: a critical review. Int. Jour. of Service Industry Management 16(1), 107–121 (2005)

[Fo97] Fowler, M.: Analysis Patterns. Reusable Object Models. Addison-Wesley (1997)

[GA95] Gailly, F., Poels, G.: Towards Ontology-driven Information Systems: Redesign and Formalization of the REA Ontology. Working Paper, Univ. Ghent (2008)

[Hoh78] Hohfeld, W.N.: Fundamental Legal Conceptions. In: Corbin (ed.), Greenwood Press, Westport (1978)

[Hr06] Hruby, P.: Model-Driven Design of Software Applications with Business Patterns. Springer (2006) ISBN: 3540301542

[Lusch08] Towards a conceptual foundation for service science: Contributions from service-dominant logic. IBM Systems Journal 47(1) (2008)

[Mc82] McCarthy W. E., The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review (1982)

[OA06] OASIS. Reference Model for Service Oriented Architecture 1.0 (2006), http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf

[Pr04] Preist, C.: A Conceptual Architecture for Semantic Web Services. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 395–409. Springer, Heidelberg (2004)

[SF06] Sampson, S.E., Froehle, C.M.: Foundation and Implications of a Proposed Unified Services Theory. Production and Operations Management 15(2) (Summer 2006)

[Sp08] Spohrer, J., et al.: The Service System Is the Basic Abstraction of Service Science. In: Proc. HICSS (2008)

[UM03] UN/CEFACT Modelling Methodology (UMM) User Guide (2003), http://www.unece.org/cefact/umm/UMM_userguide_220606.pdf (February 19, 2008)

[UN08] United Nations, Dept. of Economic and Social Affairs. Common DataBase (CDB) Data Dictionary, http://unstats.un.org/unsd/cdbmeta/gesform.asp?getitem=398 (February 19, 2008)

[VL06] Vargo, S.L., Lusch, R.F., Morgan, F.W.: Historical Perspectives on Service-Dominant Logic. In: Lusch, R.F., Vargo, S.L. (eds.) The Service-Dominant Logic of Marketing, pp. 29–42. M. E. Sharpe, Armonk (2006)

[We07] Weigand, H., et al.: Strategic Analysis Using Value Modeling–The c3-Value Approach. In: HICSS, vol. 175 (2007)

[WJAB09] Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M.: Value-Based Service Modeling and Design: Toward a Unified View of Services. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 410–424. Springer, Heidelberg (2009)

[WS04] W3C. Web Services Architecture W3C Working Group (2004), http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[Zei85] Zeithaml, V.A., Parasuraman, A., Berry, L.L.: Problems and Strategies in Services Marketing. Journal of Marketing 49, 33–46 (1985)

A Client-Centric ASM-Based Approach to Identity Management in Cloud Computing

Mircea Boris Vleju

Christian-Doppler Laboratory for Client-Centric Cloud Computing (CDCC),
Softwarepark 21, 4232 Hagenberg im Mühlkreis, Austria
b.vleju@cdcc.faw.jku.at
http://www.cdcc.faw.jku.at/

Abstract. We introduce the concept of an identity management machine (based on ASM) to mitigate problems regarding identity management in cloud computing. We decompose the client to cloud interaction into three distinct scenarios and introduce a set of ASM rules for each of them. We first consider a direct client to cloud interaction where the identity information stored on the client side is mapped to the identity created on the cloud provider's IdM system. To enhance privacy we then introduce the concept of real, obfuscated and partially obfuscated identities. Finally we take advantage of the increase in standardization in IdM systems defining the rules necessary to support authentication protocols such as OpenID. Our solution makes no supposition regarding the technologies used by the client and the cloud provider. Through abstract functions we allow for a distinct separation between the IdM system of the client and that of the cloud or service provider. Since a user is only required to authenticate once to our system, our solution represents a client centric single sign-on mechanism for the use of cloud services.

Keywords: cloud computing, abstract state machine, identity management, client centric.

1 Introduction

In today's world, any description of the successful enterprise will undoubtedly contain the words "cloud computing". Phrases such as "our cloud based solution" or "our cloud infrastructure" are a must have. The adoption of a cloud infrastructure comes with great advantages (lower costs, easier to use, more redundancy, high availability, etc.) but also has some disadvantages ([1,2,3,4]). Loss of control is its main downside, implying an extra trust level. The client now has to trust the cloud provider with the control of his data. With this loss of control, other problems follow: security and privacy issues, contracting issues, provider lock-in, etc. ([3,2]).

As such a potential client will carefully study and choose cloud providers based on the advantages/disadvantages ratio. Another key issue in adopting cloud computing is identity management. While cloud computing itself does not introduce

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 34–43, 2012.

this issue, it does broaden its impact by introducing a new way the management of identities must be performed. To further complicate the matter at hand a client can use multiple cloud providers, each having their own unique way of managing identities. Identity management suffers from some key fundamental issues such as: the definition of an identity, the single point of failure for identity providers, increased risk of phishing and likability across different domains [5,1,6]. One way to address these issues is to adopt a client centric approach.

2 Related Work

There have been several attempts to define a client centric approach to identity management. [7] describes a privacy enhanced user centric identity management system allowing users to select their credentials when responding to authentication requests. It introduces "a category-based privacy preference management for user-centric identity management" using a CardSpace compatible selector for Java and extended privacy utility functions for P3PLite and PREP languages. The advantage of such a system is that it allows users to select the specific attributes that will eventually be sent to a relying party. Such a system works well for enhancing privacy, however it fails to address the extra overhead inflicted on the user when using the system. As [8] shows a typical user would tent to ignore obvious security and privacy indicators. For composite services, [9] uses a universal identity management model focused on anonymous credentials. The model "provides the delegation of anonymous credentials and combines identity metasystem to support easy-to-use, consistent experience and transparent security".

From a client centric perspective, Microsoft introduced an identity management framework (CardSpace) aimed at reducing the reliance on passwords for Internet users authentication and at improving the privacy of information. The identity metasystem, introduced with Windows Vista and Internet Explorer 7, makes use of an "open" XML based framework allowing portability to other browsers via customized plugins. However CardSpace does suffer from some known privacy and security issues [10,11]. The concept of a client centric identity metasystem is thoroughly defined in [12]. The framework proposed here is used for the protection of privacy and the avoidance of unnecessary propagation of identity information while at the same time facilitating exchange of specific information needed by Internet systems to personalize and control access to services. By defining abstract services the framework facilitates the interoperation of the different metasystem components.

3 An ASM Based Approach

Our goal is to define a privacy enhanced client centric system based on Abstract State Machines [13]. Such a system would provide a "proxy" between the client and the cloud-based identity management systems. On an abstract level, the Identity Management Machine (IdMM) will be responsible for "translating" the protocols used by the client to manage his identities to a set of protocols used by the cloud provider. The function of this machine is to properly authenticate a user to a given cloud service as well as to manage any private identity-related data stored on the cloud.[1] The IdMM makes an abstraction of the protocols used by both the client and the provider for their identity management systems via the use of abstract functions. Such functions leave the organizational and implementation aspects the identity management systems directly into the hands of the end parties provided the appropriate functions are implemented. To define the IdMM we consider three distinct cases of client-to-cloud interaction: the direct case, the obfuscated case and the protocol based case.

3.1 Direct Client-to-Cloud Interaction

As showed by [3,6,5] one of the greatest issues surrounding cloud providers in identity management is the need of the cloud provider to control the customer experience. Many providers make use of their own custom designed identity systems to which a client must subscribe. This means that the client has no choice but to use the cloud provider identity system. While this may be an inconvenience from a privacy point of view, the real problem lies in managing the client's information across multiple providers. A simple change, such as changing a user's address, entails changing the value on every single provider the client uses. To combat this problem we view the interaction between the client and a cloud service as a bijective mapping. Any change made on the client side must also be made on the cloud.

3.2 Obfuscated Client-to-Cloud Interaction

While the direct client-to-cloud interaction allows for an efficient use of cloud services it does suffer from a lack of privacy. Since all information about a client is stored on the provider's infrastructure there is an increased risk that through data leakage or unauthorized access that information could be fall into the wrong hands. We mitigate this threat by introducing the concept of obfuscated identities. While data obfuscation is not a new technique [14], its applicability in identity management is. To introduce identity obfuscation we must first go back to the concept or definition of an identity [15,5].

Real Identity. We consider an identity as being a *real identity* if the information contained by this identity corresponds to the identity's owner and is visible to any external entity. Such identities can be used, for example, in online stores, where the provider needs to have real information to work with. In such cases the provider needs to know the full name of the user, his address and possibly credit card data. Using any kind of obfuscation in such a case could impede the provider from handling the order request.

[1] The focus of this paper is purely on identity management and not on identity and access management. Once the IdMM has been thoroughly defined it can be extended to include access rights management.

Obfuscated Identity. As opposed to a real identity, an *obfuscated identity* has its information obfuscated. Depending on the method of obfuscation [14], the information is either undecipherable or can only be deciphered by the owner of the identity. Any file storage service can be used as an example where obfuscated identities are recommended. In such services, the provider does not need to know the user's full name, height or address. Therefore such obfuscated identities can be successfully used. If the obfuscation method makes the information undecipherable, then the corresponding identity can be viewed as anonymous [9].

Partially Obfuscated Identity. A third kind of identity considered is the *partially obfuscated identity.* The information contained by such identities is a mix of real/visible attributes as well as obfuscated ones. If we extend the file storage example by adding the condition that any user must be over 18 years old in order to use the service, then the age of the user must not be obfuscated. As such, while the rest of the information can remain obfuscated, the age will contain real, unobfuscated user data.

3.3 Protocol-Based Client-to-Cloud Interaction

In recent years there has been a drive to improve interoperability between cloud providers mostly to prevent vendor lock-in [3]. From an identity management perspective the result has been the adoption of some open-based protocols to facilitate both cloud interoperability as well as identity access management. Protocols such as OpenID, OpenAuth or LDAP represent an important tool for a client centric identity management system. They allow the client to design his own system while allowing access to cloud based services via the implementation of these protocols.

3.4 Notational Conventions

For a quick reference we list here some frequently used notations, in particular for list operations:

$T*$ denotes a list of elements of type T.
$[e_1, \ldots, e_n]$ denotes a list containing the elements $e_1, \ldots, e_n$.
$[]$ denotes an empty list.
$length(l)$ returns the number of elements in the list l.
$l_1 \cdot l_2$ denotes the concatenation of the lists l_1 and l_2.
$e \in l$ denotes that the list l contains the element e.
$e \notin l$ denotes that the list l does not contain the element e.
$l_1 \subset l_2$ denotes that all the elements in l_1 exist in l_2.
$split(l, n)$ splits off the last n elements from the list l returning a pair (l', n') such that $l' \cdot n' = l \wedge length(n') = n$.
$l_1 - l_2$ removes from l_1 any elements that exist in both l_1 and l_2.
$random(l)$ returns a random element from the list l.

4 The Identity Management Machine (IdMM)

To create a privacy enhanced client-centric identity management component for cloud computing we define the identity management machine (IdMM) based on ASM, which will perform all required functionalities from authenticating the user on the machine, to automatically authenticating him to any used service.

4.1 Types and Data Structures

In order to define the IdMM we must first define a set of types and data structures used by this machine (described in figure 1). Even though our focus is on identity management we cannot leave out the access management part. The type *Access* is used to define access control lists for both users and services. Any user has a set of given attributes (name, height, age, etc.). For this we define the data structure *Attr*. The data structure *User* is used to store any information about an identity. Similarly, the data structure *Service* is used for storing information regarding services. The type of the identity (real, partially obfuscated or fully obfuscated) is given by *IdentityType*. To describe a protocol, we use the *Protocol* data structure.

$\textbf{type}\ Access\ =\ NoAccess \mid Read \mid Write \mid Execute$
$\textbf{type}\ IdentityType\ =\ Real \mid Partial \mid Obfuscated$

$\textbf{data}\ Attr\ =\ (name, value)$
$\quad name \in String\ ,\ \ value \in String \mid \mathbb{R} \mid Attr \mid Attr{*}$

$\textbf{data}\ Protocol\ =\ (name, protocolAttrs)\ ,$
$\quad name \in String\ ,\ \ protocolAttrs \in String{*}$

$\textbf{data}\ User\ =\ (id, attrs, sacl, idType)\ ,$
$\quad id \in String\ ,\ \ attrs \in Attr{*}\ ,\ \ sacl \in Map\,(Service, Access{*})\ ,$
$\quad idType \in IdentityType$

$\textbf{data}\ Service\ =\ (uri, attrs, acl, authService, authAttr, idType, protocols)$
$\quad uri \in String\ ,\ \ attrs \in String{*}\ ,\ \ acl \in Access{*}\ ,$
$\quad authService \in Service \mid \emptyset\ ,\ \ authAttr \in String{*} \mid []\ ,$
$\quad idType \in IdentityType\ ,\ \ protocols \in Protocol{*}$

Fig. 1. IdMM Types and Data Structures

4.2 States

IdMM dynamic state is represented by a single frame (figure 2) containing the current User, a list of all the authentication services the user is connected to (*logins*), a list of all the services currently used (*services*), the identity used for a particular service (*idMap*), a mapping of the protocol used (if any) to authenticate to a service (srvProtocols) and the instruction that is currently executed (*instr*).

```
data Instr = UserLogin
   | UserLogout
   | UserLogoutServices
   | Halt(String)
   | ServiceLogin(String)
   | AuthorizeLogin(String, Service)
   | PerformLogin(String, Service)
   | PerformObfuscatedLogin(String,
       Service, User)
   | PerformProtocolLogin(String, Service,
       User)
   | ServiceLogout(String)
```

```
user : User ,  user = ∅
logins : Service∗ ,  logins = []
    ∀l ∈ logins : l.authService = ∅
services : Service∗ ,  services = []
    ∀s ∈ services :
        s.authService ∈ logins∧
            s ∉ logins
instr : Instr ,  instr = UserLogin
idMap : Map(Service, User) ,
    idMap = ∅
srvProtocols : Map(Service, Protocol) ,
    srvProtocols = ∅
halt : String ,  halt = ∅
```

Fig. 2. IdMM Dynamic Frame

4.3 Rules

Figures 3 and 4 specify the execution of the IdMM via ASM rules. The IdMM halts execution when *halt* will have a defined value. As long as IdMM does not halt it fires the appropriate rule based on the value of *instr*.

User Authentication. Upon start-up the first rule fired is *UserLogin* (figure 3). This instruction prompts the user to input his credentials and authenticates him to the machine. If a user cannot be authenticated the machine halts with the appropriate error. When a user wants to log out the event triggered will set *instr* to *UserLogout*. When this rule is fired *user* is set to ϕ and *instr* is set to *UserLogoutService*. The *UserLogoutService* instruction logs the user out of every single service he was connected to and halts the execution of the machine. The *Halt* instruction causes the machine to halt execution.

Service Authentication. When the user wants to use a specific service he will specify the service's URI. The event triggered by this action will set *instr* to *ServiceLogin* (figure 4). The *ServiceLogin* instruction attempts to find the matching service given an URI. If no services can be found then an error message is triggered. If a service is found then *instr* will be set to *AuthorizeLogin*. In case the user is already connected to the service he will be redirected to the given URI. The authorization of the user to use a given service is done by the *AuthorizeLogin* instruction. Upon a successful checking of access rights *instr* is set to *PerformLogin*. This checks the type of identity required and sets the value of *instr* to *PerformObfuscatedLogin*. If the service supports protocol-based authentication then *PerformObfuscatedLogin* will set *instr* to *PerformProtocolLogin*, which will perform the authentication based on a given protocol. In case the service does not support protocol-based authentication *PerformObfuscatedLogin* will perform the authentication with the given identity. The *ServiceLogout* instruction is called when a log-out event is triggered. The instruction searches for the matching services and performs the log-out.

```
UserLogin →
   let attrs=prompt()
   let u=findUser(attrs)
   if u=∅ then
      instr:=Halt("Login failed")
   else
      user:=u
      instr:=∅
UserLogout →
   user:=∅
   instr:=UserLogoutService
UserLogoutService →
   if logins=[] then
      instr:=Halt("halt")
   else
      let (login′,s)=split(logins,1)
      performLogout(s)
      login:=login′
Halt(msg) →
   if (msg≠"halt") then
      error(msg)
   halt:=msg
```

Fig. 3. IdMM User Authentication Rules

4.4 Abstract Functions

The IdMM defines several abstract functions that allow the machine to interact with both the client and the cloud. The function *findService* searches for a service that matches the URI. The function *prompt* returns the input from the user when he is asked for the credentials. To find the identity matching these credentials we use the function *findUser*.[2] Displaing an error message is done via *error* function. We use the function *redirect* to redirect a user to a particular URI. The functions *getAuthAttrs* and *getServiceAttrs* return the authentication attributes and the service attributes for a given service. The function *setUserAttr* will replace the required attributes of a given user with the values of the current user. The function *performProtocolLogin* will return *true* upon successful authentication to the service using the provided protocol and parameters (attributes) and *false* if the authentication fails. The function *getProtocolAttributes* returns the list of protocol attributes from a specific user. The function *performLogin* is used to authenticate an identity to the given service. To syncronize the attributes on an identity with the attributes on the service the *syncServiceAttr* function is used. The function *performLogout* performs a log-out from a given authentication service.

The abstract functions represent ways to access client data and are used to retrieve or modify data from the client's IdM system. Analogously, functions such as *syncServiceAttr* or *performLogin* handle the interaction with the cloud provider's IdM system (in the direct and obfuscated cases). The protocol-based case introduces a new set of functions that implement the specific protocol used.

[2] We define the function *users()* to retrieve all users and the function *services()* to retrieve all available services

```
ServiceLogin(uri) →
    let s=findService(uri)
    if s=∅ then
        error("Service not supported")
        instr:=∅
    else
        if s∈ services ∨s.authService∈logins then
            redirect(uri)
            instr:=∅
        else
            instr:=AuthorizeLogin(uri,s)
AuthorizeLogin(uri, s) →
    if s.acl⊂user.sacl(s) then
        instr:=PerformLogin(uri,s)
    else
        error("Permission denied");
        instr:=∅
PerformLogin(uri, s) →
    case s.idType of
        Real →
            instr:=PerformObfuscatedLogin(uri,s,user)
        Partial →
            let u = random(users(s, Partial))
            setUserAttr(u, s, user)
            instr:=PerformObfuscatedLogin(uri,s,u)
        Obfuscated →
            let u = random(users(s,Obfuscated))
            instr:=PerformObfuscatedLogin(uri,s,u)
PerformObfuscatedLogin(uri, s, u) →
    if s.authService.protocols=∅ then
        if performLogin(s.authService, getAuthAttrs(s.authService, u)) then
            logins:=logins·[s.authService]
            services:=services·[s]
            idMap(s.authService):=u
            syncServiceAttr(s, getServiceAttrs(s, u))
            redirect(uri)
            instr:=∅
        else
            error("Failed authentication");
            instr:=∅
    else
        instr:=PerformProtocolLogin(uri,s,u)
PerformProtocolLogin(uri, s, u) →
    let p=random(s.authService.protocols)
    if performProtocolLogin(s, p, getProtocolAttr(u,p)·getServiceAttrs(s, u)) then
        logins:=logins·[s.authService]
        services:=services·[s]
        idMap(s.authService):=u
        srvProtocols(s.authService):=p
        redirect(uri)
        instr:=∅
    else
        error("Failed authentication");
        instr:=∅
ServiceLogout(uri) →
    let s=findService(uri)
    if s∈services then
        services:=services −[s]
        logins:=logins −[s.authService]
        if ∃ srvProtocols(s.authService) then
            performProtocolLogout(s.authService, srvProtocols(s.authService))
            srvProtocols(s.authService):=∅
        else
            performLogout(s.authService)
    instr:=∅
```

Fig. 4. IdMM Service Authentication Rules

The usage of abstract functions also makes IdMM independent from the design of the client's and provider's IdM system. As such there are many ways to implement an IdMM. For example, the IdMM can be implemented as a standalone application using a local based IdM system for the client. Similarly the client's data can be stored elsewhere and can be accessed either through existing protocols or custom designed protocols. The IdMM itself can also be implemented as a browser plugin (for services offered via web based applications) or any other application specific plugins (e.g. plugins for a specific IDE in PaaS).

5 Conclusions and Further Work

Cloud based solutions provide a client with ready-made, robust and reliable services putting focus on his priorities and eliminating the need for the maintenance of his IT infrastructure. While the advantages of cloud-based services are clear, the disadvantages can impede their adoption. From an IdM point of view, the need of the provider to control the customer experience and the lack of interoperability between providers coupled with the different client centric IdM systems present an obstruction for the adoption of a cloud based infrastructure. The introduction of the IdMM tries to simplify this problem by creating a proxy between the client and the cloud making it easier for a client to use cloud based systems. By the introduction of obfuscated access we also try to address some of the most common privacy related issues that come with the use of cloud computing.

The IdMM represents a generic description of an ASM used in managing client centric identities in a cloud based scenario. In order to ensure a correct specification of the IdMM, further refinement must be applied to each client to cloud interaction scenario. Once the specification for one on the submachines is thought to be complete a small scale proof of concept will be implemented using the required specifications. This implementation must be only sufficient to cover all the specifications listed for a particular sub-machine and will not cover all cloud based services.

The IdMM takes into consideration only three cases of client to cloud interaction. Once the specification of these three cases is complete, it can be extended by a new case which allows for the use of multiple identities to authenticate to a service. This system of routed authentication will be an extension of the protocol-based scenario. Rather than connect directly to the service, the user will connect to it using a random path of identity services which will make it harder to trace the actions of a user. For example if the targeted services uses OpenID, then the user would connect to a service which is an OpenID provider and use that service to authenticate on the targeted service. The possibilities, together with the advantages and disadvantages, of using such a system have to be studied before building the specification of a routed IdMM.

References

1. Brad, A.M.: New threats in cloud computing - with focus on identity and access management. Master's thesis, Johannes Kepler Universität Linz (July 2010)
2. Vleju, M.B.: New threats in cloud computing - with focus on cloud misuse and cloud vulnerabilities from the client side. Master's thesis, Johannes Kepler Universität Linz (July 2010)
3. Brunette, G., Mogull, R.: Security Guidance for critical areas of focus in Cloud Computing V2. 1. CSA (Cloud Security Alliance), USA (2009), http://www.cloudsecurityalliance.org/guidance/csaguide.v21
4. Fahmy, H.: New threats in cloud computing - ensuring proper connection and database forensics from the client side. Master's thesis, Johannes Kepler Universität Linz (July 2010)
5. Alpár, G., Hoepman, J.H., Siljee, J.: The identity crisis. security, privacy and usability issues in identity management. CoRR abs/1101.0427 (2011)
6. Dhamija, R., Dusseault, L.: The seven flaws of identity management: Usability and security challenges. IEEE Security Privacy 6(2), 24–29 (2008)
7. Ahn, G.J., Ko, M., Shehab, M.: Privacy-enhanced user-centric identity management. In: IEEE International Conference on Communications, ICC 2009, pp. 1–5 (June 2009)
8. Schechter, S., Dhamija, R., Ozment, A., Fischer, I.: The emperor's new security indicators. In: IEEE Symposium on Security and Privacy, SP 2007, pp. 51–65 (May 2007)
9. Zhang, Y., Chen, J.L.: Universal identity management model based on anonymous credentials. In: 2010 IEEE International Conference on Services Computing (SCC), pp. 305–312 (July 2010)
10. Alrodhan, W., Mitchell, C.: Addressing privacy issues in cardspace. In: Third International Symposium on Information Assurance and Security, IAS 2007, pp. 285–291 (August 2007)
11. Oppliger, R., Gajek, S., Hauser, R.: Security of microsoft's identity metasystem and cardspace. In: Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference, February 26 - March 2, pp. 1–12 (2007)
12. Cameron, K., Posch, R., Rannenberg, K.: Proposal for a Common Identity Framework: A User-Centric Identity Metasystem (2008)
13. Börger, E., Stärk, R.F.: Abstract State Machines. A Method for High-Level System Design and Analysis. Springer (2003)
14. Bakken, D., Rarameswaran, R., Blough, D., Franz, A., Palmer, T.: Data obfuscation: anonymity and desensitization of usable data sets. IEEE Security Privacy 2(6), 34–41 (2004)
15. The Open Group Identity Management Work Area: Identity management. White Paper (March 2004)

International Workshop on Evolution and Change in Data Management and on Non Conventional Data Access (ECDM – NoCoDa 2012)

Preface

Traditionally, the database research community has focused on methodologies, techniques, and technologies for data management to support and enable business activities. Yet, in the last couple of decades this equilibrium has been completely overturned, as data-management research and development problems became part of every individual and collective activity in our society. New ways of representing information require equally new approaches and technologies for its storage and processing: examples of these range from multimedia data streams and storage systems to semantic-web knowledge linked all over the Web, and from scientific time series to natural language information that has to be understood by software programs. Therefore, a data-centric vision of the world is actually key for advancing in such a demanding scenario, where two orthogonal phenomena take place: on the one hand, rather than storing, managing and analyzing only the current state of the information we are forced to deal with the *management of information change*; on the other hand, the *new ways to access data* must take into consideration the kind of data available today and a new population of prospective users.

In this framework, the ECDM–NoCoDa Workshop addresses the following two problems, bringing together researchers and practitioners from the more established research areas as well as from emerging, visionary ones.

- Evolution and Change in Data Management (ECDM): change is a fundamental but sometimes neglected aspect of information management. The management of evolution and change and the ability for data and knowledge-based systems to deal with change is an essential component in developing and maintaining truly useful systems that minimize service disruption and down time and maximize availability of data and applications. Many approaches to handling evolution and change have been proposed in various areas of data management and this workshop will deal with the manner in which change can be handled, and the semantics of evolving data, metadata and their structure in computer based systems.
- Non Conventional Data Access (NoCoDa): as more and more data become available to a growing multitude of people, the ways to access them are rapidly evolving, originating a steadily growing set of proposals of non-conventional ways for data access while inheriting, where possible, the formidable equipment of methods, techniques and methodologies that have been

> produced during the last forty years. These new proposals embrace the new challenges, suggesting fresh approaches to data access that rethink the traditional information access methods in which queries are posed against a known and rigid schema over a structured database. This workshop will contribute advances on the conceptual and semantic aspects of non-conventional methods for data access and on their practical application to modern data and knowledge management.

As a result of the calls for papers, an international and highly-qualified program committee — assembled from universities and research centers worldwide — received 10 submissions overall, and after rigorous refereeing eventually chose 4 high-quality papers for presentation at the workshop and publication in these proceedings. We would like to express our thanks to the program committee members and the additional external referees for their timely expertise in reviewing, to the authors for submitting their papers, and to the ER 2012 organizers for their support.

Besides the selected paper presentations, the ECDM–NoCoDA 2012 workshop program is enriched by a keynote speech given by Paolo Terenziani titled "The Telic/Atelic Distinction in Temporal Databases" and by the panel "The relational model is dead, SQL is dead, and I don't feel so good myself", a playful though effective way to convey the dramatic change that database research on data access is recently undergoing.

October 2012

Fabio Grandi
Giorgio Orsi
Letizia Tanca
Riccardo Torlone

The Telic/Atelic Distinction in Temporal Databases

Paolo Terenziani

Institute of Computer Science, DISIT, Univ. Piemonte Orientale “A. Avogadro”,
Viale Teresa Michel 11, Alessandria, Italy
terenz@mfn.unipmn.it

Abstract. Prior research in philosophy, linguistics, artificial intelligence and other areas suggests the need to differentiate between temporal facts with goal-related semantics (i.e., telic) from those that are intrinsically devoid of culmination (i.e., atelic). We investigate the impact of the telic/atelic distinction on temporal databases, considering data and query semantics, query languages, and conceptual models.

1 Background

The telic/atelic distinction has a long tradition in the Western culture. Aristotle, in [4], first pointed out that the facts that can happen in the world can be subdivided into two main classes: *telic* facts, i.e., facts that have a specific goal or culmination (telos means “goal” in ancient Greek) and atelic facts, i.e., facts that do not have it (“a” stands for privative “α” in Greek). Later on, the telic/atelic dichotomy has been studied in many areas, and, in particular, in philosophy (ontology) and linguistics.

The telic/atelic distinction seems to play a fundamental role in human cultures. For instance, some approaches in cognitive science have pointed out that the *aktionsart* distinctions (and, in particular, the *telic/atelic* distinction) play a fundamental role in the acquisition of verbal paradigms by children (see, e.g., [6] as regards English, [8] for French, [1] for Turkish).

The linguistic community agrees that natural language sentences can be classified within different *aktionsart* classes (e.g., *activities*, *accomplishment*, *achievements* and *states* in [27]; also called *aspectual classes* [28]) depending on their linguistic behavior or on their semantic properties. These semantic properties demonstrate that *the semantics of the association of facts with time depends on the classes of facts being considered.* For example [12] has proposed the following semantic criteria to distinguish between states and accomplishments.

(1) *A sentence φ is stative if it follows from the truth of φ at an interval I that φ is true at all subintervals of I (e.g., if John was asleep from 1:00 to 2:00 PM, then he was asleep at all subintervals of this interval: be asleep is a stative).*

(2) *A sentence φ is an accomplishment/achievement (or kinesis) if it follows from the truth of φ at an interval I that φ is false at all subintervals of I (e.g., if John built a house in exactly the interval from September 1 until June 1, then it is false that he built a house in any subinterval of this interval: building a house is an accomplishment/achievement)* [12].

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 47–56, 2012.

Property (1) above has been often called ***downward inheritance*** in the Artificial Intelligence literature (e.g. [19]). Notice that also ***upward inheritance*** [19] holds over states: if John was asleep from1:00 to 2:00 and from 2:00 to 3:00, then he was asleep from 1:00 to 3:00. *States* (as well as *activities*, such as "*John is running*") are *atelic*: they denote "*homogenous*" situations without any goal or culmination, so that both downward and upward inheritance holds on them. *Accomplishments* are *telic*: they denote situations in which a culmination/goal has to be reached, so that neither downward nor upward inheritance holds on them.

The linguistic community agrees that, although all *base* facts can be classified as telic/atelic, a telic-to-atelic (or atelic-to-telic) coercion can always be performed using explicit linguistic tools; e.g., a telic sentence can be converted into atelic by applying a progressive form to strip out its culmination [18]. For example, "*Bob built a house from June 1 to September 1*" is telic and one cannot infer that "*Bob built a house on July 1*". However, one can correctly assert that "*Bob was building a house on July 1*" since a progressive form has been used in the telic-to-atelic coercion.

Starting from the pioneering work in [5], several linguistic approaches have pointed out that the traditional *point-based semantics*, in which facts are evaluated at each time point, properly applies only to atelic facts, while a *period-based semantics* (called *interval*-based by the linguistic literature) is needed to cope with telic facts.

Since "*one of the most crucial problems in any computer system that involves representing the world is the representation of time*" [3], the treatment of the telic/atelic dichotomy has had a significant impact on many areas of Computer Science, including, e.g., Artificial Intelligence (AI). In AI, the telic/atelic dichotomy (using a different terminology) was first explicitly dealt with in Allen's reified logic [2]. Many recent AI approaches concerning formal ontologies pay specific attention to the telic/atelic dichotomy.

On the other hand, though temporal databases model the ***validity time*** of facts (*transaction time* is not interesting when considering the telic/atelic issue) the treatment of the telic/atelic dichotomy has been considered by the database community only very recently, starting from the pioneering work in [22]. This is a major drawback since "*effective exchange of information between people and machines is easier if the data structures that are used to organize the information in the machine correspond in a natural way to the conceptual structures people use to organize the same information*" [18, p. 26]. Though previous research in philosophy and linguistics has identified many different features to characterize the telic/atelic distinction, considering Temporal Databases (TDBs) we can adopt the following simplified definition (see also [Kathri et al., 2009]):

Definition (Telic/atelic facts). *Atelic facts* (*data*) are facts (data) for which both *downward* and *upward inheritance* hold; *Telic facts* are facts for which neither *downward* nor *upward inheritance* hold.

In the following, we explore the impact of the telic/atelic distinction on TDBs, focusing on the relational model, and considering data and query semantics, and, briefly, query languages and conceptual models. To understand the dissertation, it is important to keep in mind three distinctions:

(1) Representation versus semantics of the language—Our approach concerns the temporal *semantics* of data and queries, independent of the *representation* one uses for time. This distinction is analogous to the distinction between *concrete* and *abstract* databases in [9].

(2) Data language versus query language—The two should be differentiated [10]. For instance, ATSQL2 [7], SQL/Temporal [21], and SQL/TP [25, 26] support time periods in their data representation language; however, while the query languages of ATSQL2 [11] and SQL/Temporal are based on time periods, that of SQL/TP is based on time points.

(3) Data semantics versus query semantics—In most database approaches, the semantics of data is not distinguished from the semantics of the query. On the other hand, data have their own semantics, independently of any query language. This is the usual approach in AI and logics: Logical formulæ have an intrinsic meaning, which can be formally defined in model-theoretic terms. Queries are an operational way of making such a semantics explicit. However, a set of logical formulæ has a semantics per se, even if no query is asked. Analogously, we will say that data in a database have a semantics, which we will call "semantics for data".

2 Data Semantics

Though there is quite a variety of temporal relational database approaches in general, and data models in particular, most of them have a common underlying feature: they assume a point-based semantics for data (Section 2.1). Unfortunately, as pointed out by the linguistic research, point-based semantics is inadequate for telic facts (Section 2.2), for which a more expressive period-based (called *interval-based* in linguistics) semantics (Section 2.3) is needed.

2.1 Point-Based Data Semantics

In the point-based semantics the data in a temporal relation is interpreted as a sequence of states (with each state a conventional relation: a set of tuples) *indexed by points in time*. Each state is independent of every other state.

Such temporal relations can be encoded in many different ways (data language). For example the following are three different encodings of the same information, within a point-based semantics, of John being married to Mary in the states indexed by the times 1, 2, 7, 8, and 9:

(i) <John, Mary || {1,2,7,8,9}> ∈ R
(ii) < John, Mary || {[1–2],[7–9]}}> ∈ R
(iii) < John, Mary || [1–2]> ∈ R and <John, Mary || [7–9]> ∈ R

Independently of the representation, the point-based semantics is that the fact denoted by < John, Mary > is in 5 individual states, as follows.

1 → {< John, Mary >} 2 → {< John, Mary >} 7 → {< John, Mary >}
8 → {< John, Mary >} 9 → {< John, Mary >}

It is worth stressing that, despite many differences due to different implementation and/or representation strategies, and despite the fact that several query and data representation languages include time periods, all TDBs approaches adopt, explicitly or implicitly, the point-based semantics (consider, e.g., BCDM [15], the "consensus" semantic model that has been identified to isolate the "common semantic core" of TSQL2 and many other TDB approaches). Adopting the point-based semantics grants that a temporal database can be interpreted as a set of non-temporal databases, one at each point in time. This interpretation has theoretical and practical advantages: Properties such as *upward compatibility* and the *reducibility* of temporal algebrae to SQL [7] can be obtained, granting for the *interoperability* of temporal approaches with non-temporal databases, and their *implementability* on top of them. Indeed, point-based semantics also naturally support inheritance properties, and thus is adequate to deal with atelic facts.

Note (Point-based semantics and inheritance properties). Notice that the point-based semantics naturally applies to atelic facts, since both downward and upward inheritance are naturally supported.

Unfortunately, upward and downward inheritance do not hold for telic facts. This fact arises doubts about the adequacy of point-based semantics to deal with them.

2.2 Inadequacy of Point-Based Semantics to Deal with Telic Data

As an example of telic facts, let us consider drug administration. For example, let us suppose that Sue had an administration of 500 mg of cyclophosphamide (a cancer drug) starting at 1 and ending at 3 (inclusive), an administration of 500 mg of cyclophosphamide starting at 4 and ending at 4, and that Mary had an administration of 200 mg of infliximab (a monoclonal antibody) starting at 2 and ending at 4. If we adopt the point-based semantics, the above example is modeled as:

1 → {<Sue, cyclophosphamide, 500>}
2 → {<Sue, cyclophosphamide, 500>, <Mary, infliximab, 200>}
3 → {<Sue, cyclophosphamide, 500>, <Mary, infliximab, 200>}
4 → {<Sue, cyclophosphamide, 500>, <Mary, infliximab, 200>}

As predicted by the linguistic literature, *Point-based semantics is not expressive enough* to deal with the temporal interpretation of *telic data*, so that, in the example, we have a loss of information. The two different drug administrations given to Sue (one starting at 1 and ending at 3, the other from 4 to 4) cannot be distinguished in the semantic model. The effects of such a loss of information are quite critical. For instance, in the representations above, there is no way to recover the fact that 1000 ml of cyclophosphamide in total were administered to Sue.

Note that such a loss of information is completely independent of the representation language used to model data. For instance, the above example could be represented as

(i) <Sue, cyclophosphamide, 500 || {1,2,3,4}> ∈ R and
<Mary, infliximab, 200 || {2,3,4}> ∈ R
(ii) <Sue, cyclophosphamide, 500 || {[1–3],[4–4]}> ∈ R and
<Mary, infliximab, 200 || {[2–4]}> ∈ R
(iii) <Sue, cyclophosphamide, 500 || [1–3]> ∈ R and
<Sue, cyclophosphamide, 500 || [4–4]> ∈ R and
<Mary, infliximab, 200 || {[2–4]}> ∈ R

But, as long as the point-based semantics is used, the data semantics is the one elicited above (formally, the representations (i)-(iii) above are *snapshot equivalent* [15]). This is true for all approaches that use *temporal elements* to timestamp tuples; consider SQL/Temporal, TSQL2, TSQL, HQL, and TQuel, which utilize temporal elements to timestamp tuples, and Gadia's [13] Homogeneous Relational Model, which uses temporal elements to timestamp attributes. While temporal elements are sets of time periods, this is only a matter of data representation language since the underlying data semantics is point-based (i.e., in such approaches, temporal elements are merely a notational representation for a set of time points). Thus, the loss of information is the same in all the representations. As predicted by the linguistic literature, regardless of the chosen representation, a more expressive semantics, *Period-based semantics*, is needed to cope properly with telic data!

2.3 Period-Based Data Semantics

In the Period-based semantics each tuple in a temporal relation is associated with a set of time periods, which are the temporal extents in which the fact described by the tuple occur. In this semantics *the index is a time period*. Time periods are *atomic primitive entities*, in the sense that they cannot be decomposed. Note, however, that time periods can overlap; there is no total order on time periods, unlike time points.

For instance, in the period-based semantics, the drug administration example can be modelled as follows:

[1–3] → {<Sue, cyclophosphamide, 500>}
[2–4] → {<Mary, infliximab, 200>}
[4–4] → {<Sue, cyclophosphamide, 500>}

Note that if a period-based semantics is used, the above information does not imply that Sue had a cyclophosphamide administration of 500 mg at 1, or one in the period [1–4]. In period-based semantics, periods are atomic entities, that cannot be merged or decomposed. This fact correctly accounts for the fact that neither downward nor upward inheritance hold for telic facts.

Note (Period-based semantics and inheritance properties). Notice that the period-based semantics naturally applies to telic facts, since neither downward not upward inheritance are supported.

To wrap up, point-based data semantics is needed to deal with atelic data, and period-based semantics is required for telic ones. A two-sorted semantic approach to valid-time relational data has been first provided, in the area of temporal databases, by Terenziani and Snodgrass [23]. In such an approach, both atelic relations (i.e., relations based on the point-based semantics) and telic relations (i.e., relations based on the period-based semantics) are supported.

3 Query Semantics

Results of queries should depend only on the data semantics, not on the data representation. As a consequence, the problems due to the treatment of telic data in a point-based (atelic) framework shown in Section 2.2 above are even more evident when queries are considered. For instance, considering the drug administration example above (and independently of the chosen representation), queries about the number or durations of administrations would not provide the desired answers if the point-based semantics is assumed. For instance, the number of drug administrations to Sue would be one, and an administration would (incorrectly) be provided to a query asking for administrations lasting for at least 4 consecutive time units. Dual problems arise when querying telic data coped with in an atelic (Point-based) framework.

In order to cope with a data model supporting both telic and atelic relations, temporal query languages must be extended. Specifically, queries must cope with atelic relations, telic relations, or a combination of both.

Furthermore, linguistic research points out that, while basic facts can be classified as telic or atelic, natural languages provides several ways to switch between the two classes. For the sake of expressiveness, it is desirable that a database query language provides the same flexibility.

3.1 Queries about Atelic Data

Most temporal database approaches are based (implicitly or explicitly) on the Point-based semantics. Thus, the corresponding algebraic operators already cope with atelic data. For instance, in BCDM, the union of two relations is obtained by taking the tuples of both relations, and “merging” *value equivalent* tuples, performing the union of the time points in their valid time. This definition is perfectly consistent with the “point-by-point” view enforced by the underlying point-based (atelic) semantics.

Interestingly, many algebræ in the literature also contain operators which contrast with such a “point-by-point” underlying semantics. Typical examples are *temporal selection* operators. For instance, whenever a duration is asked for (e.g., “retrieve all persons married for at least *n* consecutive time units”), the query implicitly assume a telic interpretation of data, in which time points are not taken into account independently of each others, but the duration of periods covering them is considered.

3.2 Queries about Telic Data

Algebraic operators on telic data can be easily defined as a polymorphic variant of the atelic definitions, considering that, in the telic case, the basic temporal primitives are not time points, but time periods [23]. For instance, telic union is similar to the atelic one, except that the merging of valid times of value-equivalent tuples is performed by making the union of sets of time periods, considered as primitive entities, e.g.,

$$\{[10–12],[13–15]\} \cup \{[10–14],[13–18]\} = \{[10–12],[13–15],[10–14],[13–18]\}$$

Note that such temporal selection operators perfectly fit with the telic environment. On the other hand, algebraic operators that intuitively involve a point-by-point view of data (e.g., Cartesian product, intuitively involving a point-by-point intersection between valid times) have an awkward interpretation in the telic context.

3.3 Queries Combining Telic and Atelic Data

In general, if a two-sorted data model is used, queries combining relations of both kinds are needed. In general, such queries involve the (explicit or implicit) coercion of some of the relations, to make the sort of the relations consistent with the types of the operators being used. For instance, the following query utilizes the example atelic relation modeling marriages and the telic one considering drug administrations: "*Who were being married when Sue was having a drug administration?*". In such a case, the English clause "*when*" demands for an atelic interpretation: the result can be obtained by first coercing the drug-administration relation into an atelic relation, and then by getting the temporal intersection through the application of the atelic Cartesian product. Interestingly, the very same relation may be interpreted both as telic and atelic within the same query. Consider, for instance, the query: "*Who had a (complete) drug administration during the time when Sue was having drug administrations*?". The progressive form in "*was having drug administrations*" demands from a coercion to atelic of the drug administration relation. As a result, the two adjacent administrations to Sue in the example are *coalesced* into a unique one. After that, the application of the temporal predicate "*during*" demands for a new coercion into telic. As a result, Mary's administration is (correctly) obtained in the example, even if it is not contained in the time of any specific drug administration to Sue.

In general, two coercion operators need to be provided [23]. Coercion from telic to atelic is easy: each time period constituting the (semantics of the) valid time is converted into the set of time points it contains (e.g., to-atelic({[1–3],[4–4]}) = {1,2,3,4}). Of course, since sets of time periods are more expressive than sets of time points, such a conversion causes a loss of information. On the other hand, coercion from atelic to telic demands the formation of time periods out of sets of points: the output is the set of maximal convex time periods exactly covering the input set of time points, e.g., to-telic({1,2,3,4}) = {[1–4]}.

A temporal relational algebra coping with all the above issues has been provided in [23].

4 Query and Data Languages

In [24], the authors show how the above concepts can be added to an SQL-based temporal query language. Only a few new constructs are needed. The specifics (such as using TSQL2) are not as important; the core message is that incorporating the distinction between telic and atelic data into a user-oriented query language is not difficult.

In order to limit the changes to TSQL2 (which is an atelic framework), [24] impose that, if no explicit indication is provided, temporal data are atelic. Thus, the first extension is to support the definition of telic tables (the default is designated as atelic). This can be done with an "**`AS TELIC`**" clause in the TSQL2 **`CREATE TABLE`** statement. By default, the result of queries is an atelic relation; Thus, for telic queries, Terenziani et al. prepend the keyword "TELIC" (i.e., "TELIC SELECT …"). The last extension to TSQL2 regards the need, in the queries, to adopt *coercion* functions to convent tables of the different sorts. Thus, coercion functions

`TELIC` and **`ATELIC`** are introduced.

5 Conceptual models

Given its relevance, it is important that the telic/atelic distinction can also be captured during the conceptual design of TDBs. Many different approaches have extended non-temporal conceptual data models to cope with time (see, e.g., the survey [14]). Instead of proposing yet another temporal conceptual model, in order to differentiate between telic and atelic data semantics in conceptual database design, Khatri et al. [17] propose an annotation-based temporal conceptual model that generalizes the semantics of conventional conceptual models. Their temporal conceptual design approach involves: 1) capturing "what" semantics using a conventional conceptual model; 2) employing annotations to differentiate between telic and atelic data semantics that help capture "when" semantics; 3) specifying temporal constraints, specifically *non-sequenced constraints*, in the temporal data dictionary as metadata. Khatri et al.'s approach provides a mechanism to represent telic/atelic temporal semantics using temporal annotations. They also show how this semantics can be formally defined using constructs of the conventional conceptual models and axioms in first-order logic. Via the semantics implied by the interaction of annotations, they illustrate the logical consequences of representing telic/atelic data semantics during temporal conceptual design.

6 Conclusions

Until the pioneering work in [22], all TDBs approaches have been based, explicitly or implicitly, on the "*Point-based*" (or "*snapshot*") semantics, dictating that a temporal database must be interpreted as a set of conventional (non-temporal) databases, one at each point (snapshot) of time. However, the *telic/atelic* dichotomy, dating back to Aristotle's *Categories*, dictates that *telic facts have a different ("Period-based") semantics*. As a consequence, we show that past approaches *cannot deal correctly with telic facts*, and revisit and extend the whole apparatus of current TDBs (data semantics, query semantics, query languages, conceptual formalisms) to overcome such a major drawback.

Acknowledgements. The research activity about the telic/atelic dichotomy in temporal databases described in this paper has been carried on by Paolo Terenziani in strict cooperation with Richard T. Snodgrass. The author is also very grateful to Alessio Bottrighi, Mauro Torchio and Gianpaolo Molino, who contributed to the definition of the query and data languages (sketched in Section 4), and to Vijay Khatri and Sudha Ram, who contributed to the extension of the temporal conceptual model (Section 5).

References

1. Aksu, A.: Aspect and Modality in the Child's acquisition of the Turkish Past Tense. Dissertation. Univ. of California at Berkeley (1978)
2. Allen, J.F.: Toward a General Theory of Action and Time. Artificial Intelligence 23(2), 123–154 (1984)
3. Allen, J.F.: Time and time again: The many ways to represent time. International Journal of Intelligent Systems 6(4), 341–356 (1991)
4. Aristotle: The Categories, on Interpretation. Prior Analytics. Harvard University Press, Cambridge
5. Bennet, M., Partee, B.: Tense and Discourse Location In Situation Semantics. Indiana University Linguistics Club, Bloomington (1978)
6. Bloom, L., Lifter, L., Hafitz, J.: Semantics of Verbs and the Developments of Verb Inflection in Child Language. Language 52(2), 386–412 (1980)
7. Bohlen, M., Jensen, C.S., Snodgrass, R.T.: Temporal Statement Modifiers. ACM Trans. Database Systems 25(4), 407–456 (2000)
8. Bronckart, J.P., Sinclair, H.: Time, tense and aspect. Cognition 2, 107–130 (1973)
9. Chomicki, J.: Temporal Query Languages: A Survey. In: Gabbay, D.M., Ohlbach, H.J. (eds.) ICTL 1994. LNCS, vol. 827, pp. 506–534. Springer, Heidelberg (1994)
10. Chomicki, J., Toman, D.: Temporal Logic in Information Systems. In: Chomicki, J., Saake, G. (eds.) Logics for Databases and Information Systems, ch.3, pp. 31–70 (1998)
11. Chomicki, J., Toman, D., Bohlen, M.H.: Querying ATSQL Databases with Temporal Logic. ACM Trans. Database Systems 26(2), 145–178 (2001)
12. Dowty, D.: The Effects of the Aspectual Class on the Temporal Structure of Discourse, Tense and Aspect in Discourse. Linguistics and Philosophy 9(1), 37–61 (1986)
13. Gadia, S.K.: A Homogeneous Relational Model and Query Languages for Temporal Databases. ACM Trans. Database Systems 13(4), 418–448 (1988)
14. Gregersen, H., Jensen, C.S.: Temporal Entity-Relationship Models-A Survey. IEEE Transactions on Knowledge and Data Engineering 11(3), 464–497 (1999)
15. Jensen, C.S., Snodgrass, R.T.: Semantics of Time-Varying Information. Information Systems 21(4), 311–352 (1996)
16. Khatri, V., Snodgrass, R.T., Terenziani, P.: Telic distinction in temporal databases. In: Liu, L., Tamer Özsu, M. (eds.) Encyclopedia of Database Systems. Springer (2009)
17. Khatri, V., Ram, S., Snodgrass, R.T., Terenziani, P.: Capturing Telic/Atelic Temporal Data Semantics: Generalizing Conventional Conceptual Models. IEEE Transaction on Knowledge and Data Engineering (accepted paper, 2012)
18. Moens, M., Steedman, M.: Temporal Ontology and Temporal Reference. Computational Linguistics 14(2), 15–28 (1988)
19. Shoham, Y.: Temporal logics in Al: semantical and ontological considerations. Artificial Intelligence 33(1), 89–104 (1987)

20. Snodgrass, R.T. (ed.): The Temporal Query Language TSQL2. Kluwer Academic Pub. (1995)
21. Snodgrass, R.T., Böhlen, M.H., Jensen, C.S., Steiner, A.: Transitioning Temporal Support in TSQL2 to SQL3. In: Etzion, O., Jajodia, S., Sripada, S. (eds.) Temporal Databases: Research and Practice. LNCS, vol. 1399, pp. 150–194. Springer, Heidelberg (1998)
22. Terenziani, P.: Is point-based semantics always adequate for temporal databases? In: Proc. TIME 2000, Cape Breton, Canada, pp. 191–199. IEEE Press (2000)
23. Terenziani, P., Snodgrass, R.T.: Reconciling Point-based and Interval-based Semantics in Temporal Relational Databases: A Treatment of the Telic/Atelic Distinction. IEEE Transactions on Knowledge and Data Engineering 16(5), 540–551 (2004)
24. Terenziani, P., Snodgrass, R.T., Bottrighi, A., Molino, G., Torchio, M.: Extending Temporal Databases to Deal with Telic/Atelic Medical Data. Artificial Intelligence in Medicine 39(2), 113–126 (2007)
25. Toman, D.: Point vs. Interval-Based Query Languages for Temporal Databases. In: Proc. ACM Symp. Principles of Database Systems, pp. 58–67 (1996)
26. Toman, D.: Point-Based Temporal Extensions of SQL and Their Efficient Implementation. In: Etzion, O., Jajodia, S., Sripada, S. (eds.) Temporal Databases: Research and Practice. LNCS, vol. 1399, pp. 211–237. Springer, Heidelberg (1998)
27. Vendler, Z.: Verbs and times. Linguistics in Philosophy, pp. 97–121. Cornell University Press (1967)
28. Webber, B.: Tense and aspect. Computational Linguistics 2(14) (1988)

Static Analysis of XML Document Adaptations

Alessandro Solimando, Giorgio Delzanno, and Giovanna Guerrini

DIBRIS, Università di Genova, Italy
{alessandro.solimando,delzanno,guerrini}@unige.it

Abstract. In this paper we propose a framework for XML data and schema coevolution that allows to check whether a user-proposed document *adaptation* (i.e., a sequence of document update operations intended to adapt the documents valid for a schema to a new schema) is guaranteed to produce a document valid for the updated schema. The proposed framework can statically determine, working only with the automata related to the original and modified schema, if the document update operation sequence will re-establish document validity, thus avoiding the very expensive run-time revalidation of the set of involved documents that is usually performed upon schema update.

1 Introduction

Practical data management scenarios are characterized by an increasing dynamicity and rapid evolution so that updates to data, as well as to their structures, are very frequent. Only an efficient support for changes to both data and structural definitions can guarantee an appropriate use of schemas [12,10]. Indeed schema updates have a strong impact on existing data. Specifically, the term co-evolution refers to the ability of managing the mutual implications of data, schema, and application changes.

This need is extremely pressing in the context of the eXtensible Markup Language (XML), which has become in the last ten years a standard for data representation and exchange and is typically employed in highly evolving environments. Upon any update at schema level, XML documents valid for the original schema are no longer guaranteed to meet the constraints described by the modified schema and might need to be *adapted* [8,2]. An automatic adaptation of associated documents to the new schema definition is possible in some cases, but it is not always obvious how to re-establish document validity broken by a schema update. More flexible, user-defined adaptations, corresponding to arbitrary document update statements, would allow the user to specify, for instance through XQuery Update Facility (XQUF) [17] expressions, ad hoc ways to convert documents valid for the old schema in documents valid for the new schema. These arbitrary, user-defined adaptations, however, are not guaranteed to produce documents valid for the new schema and dynamic revalidation is needed, which may be very expensive for large document collections.

In this paper, we propose a static analysis framework (*Schema Update Framework*) for a subset of XQUF based on *Hedge Automata*, a symbolic representation of infinite sets of XML documents given via unranked trees. Specifically, the framework allows to check whether a user-proposed document *adaptation* (i.e., a set of document update

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 57–66, 2012.

operations intended to adapt the documents related to a schema that have just been updated by a known sequence of schema update operations) is guaranteed to produce a document valid for the updated schema. The framework relies on a transformation algorithm for Hedge Automata that captures the semantic of document update operations. The transformation rules allow to reason about the impact of user-defined adaptations, potentially avoiding run-time revalidation for safe adaptations.

The proposed framework can statically determine, working only with the automata related to the original and modified schema, if the document update operation sequence will preserve the validity of the documents. If so, every document valid w.r.t. the original schema, after the application of the document update sequence, will result in a new modified document valid w.r.t. the updated schema. Thus, run-time revalidation of the set of involved documents can be avoided.

The remainder of the paper is organized as follows: Section 2 introduces the preliminaries needed in the following sections, Section 3 illustrates the proposed framework, finally, in Section 4 we discuss related work, Section 5 concludes the work.

2 Preliminaries

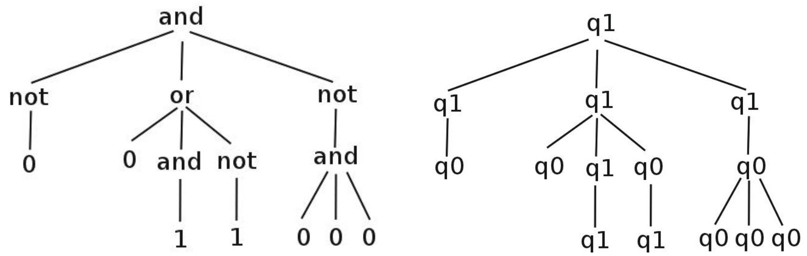
In this section we introduce Hedge Automata, as a suitable formal tool for reasoning on a representation of XML documents via unranked trees, and the update operations our framework relies on.

Hedge Automata (HA). Tree Automata are a natural generalization of finite-state automata that one can adopt to define languages over ranked finite trees. Tree Automata are used as a formal support for XML document validation. In this setting, however, it is often more convenient to consider more general classes of automata, like Hedge and Sheaves Automata, to manipulate both ranked and unranked trees. The difference between ranked and unranked trees lies in the arity of the symbols used as labels. Ranked symbols have fixed arities, that determine the branching factor of nodes labelled by them. Unranked symbols have no such a constraint. Since in XML the number of children of a node with a certain label is not fixed a priori, and different nodes sharing the same label can have a different number of children, unranked trees are more adequate for XML.

Given an unranked tree $a(t_1, \ldots, t_n)$ where $n \geq 0$, the sequence $t_1, \ldots, t_n$ is called **hedge**. For $n = 0$ we have an empty sequence, represented by the symbol ϵ. The set of hedges over Σ is $H(\Sigma)$. Hedges over Σ are inductively defined as follows: the empty sequence ϵ is a hedge; if g is a hedge and $a \in \Sigma$, then $a(g)$ is a hedge; if g and h are hedges, then gh is a hedge.

Example 1. *Given the tree $t = a(b(a, c(b)), c, a(a, c))$, the corresponding hedges having as root nodes the children of the root of t are $b(a(c(b)))$, c and $a(a, c)$.* □

A **Nondeterministic Finite Hedge Automaton** (NFHA) defined over Σ is a tuple $M = (Q, \Sigma, Q_f, \Delta)$ where Σ is a finite and non empty alphabet, Q is a finite set of states, $Q_f \subseteq Q$ is the set of final states, also called accepting states, Δ is a finite set of transition rules of the form $a(R) \rightarrow q$, where $a \in \Sigma$ and $R \subseteq Q^*$ is a regular language



(a) Tree t representing a true Boolean formula.

(b) Accepting computation of the automaton M over tree t.

Fig. 1. An example of tree t (left) and the computation $M||t$ of the automaton M over t (right)

over Q, $q \in Q$. Regular languages, denoted with R, that appear in rules belonging to Δ are said **horizontal languages** and represented with Nondeterministic Finite Automata (NFA). The use of regular languages allows us to consider unranked trees. For instance, $a(q^*)$ matches a node a with any number of subtrees generated by state q.

A **computation** of M over a tree $t \in T(\Sigma)$ corresponds to a bottom-up visit of t during which node labels are rewritten into states. More precisely, consider a node with label a such that the root nodes of its children have been rewritten into the list of states $q_1 \dots q_n \in Q$. Now, if a rule $a(R) \to q$ exists with $q_1 \dots q_n \in R$, then a can be rewritten into q, and so on.

A tree t is said to be **accepted** if there exists a computation in which the root node is labelled by $q \in Q_f$. The **accepted language** for an automaton M, denoted as $L(M) \subseteq T(\Sigma)$, is the set of all the trees accepted by M.

Example 2 (from [4]). *Consider the NFHA* $M = (Q, \Sigma, Q_f, \Delta)$ *where* $Q = \{q_0, q_1\}$, $\Sigma = \{0, 1, not, and, or\}$, $Q_f = \{q_1\}$ *and* $\Delta = \{not(q_0) \to q_1, not(q_1) \to q_0, 1(\epsilon) \to q_1, 0(\epsilon) \to q_0, and(Q^*q_0Q^*QQ^*) \to q_0, and(q_1q_1^*q_1) \to q_1, or(Q^*q_1Q^*QQ^*) \to q_1, or(q_0q_0^*q_0) \to q_0\}$. *Fig. 1(a) and Fig. 1(b) show the tree* t *representing a Boolean formula and the accepting computation of the automaton* M *(i.e.,* $M||t(\epsilon) = q_1 \in Q_f$*), respectively.* □

Given two NFHAs M_1 and M_2, the *inclusion test* consists in checking whether $L(M_1) \subseteq L(M_2)$. It can be reduced to the emptiness test for HA ($L(M_1) \subseteq L(M_2) \Leftrightarrow L(M_1) \cap (T(\Sigma) \setminus L(M_2)) = \emptyset$). Inclusion is decidable, since complement, intersection and emptiness of HA can be executed algorithmically [4].

XQuery Update Facility Operations as Parallel Rewriting. XQuery Update Facility (XQUF) [17] is a W3C recommendation as update language for XML. Its expressions are converted into an intermediate format called Pending Update List, that uses the primitives shown in the first column of Table 1. In the second column of Table 1, the tree rewriting rule corresponding to each primitive is shown. In the rules, a and b are labels, and p is an automaton state that can be viewed as a type declaration (defining any tree accepted by state p). For instance, consider the rule INS_{first} defined as $a(X) \to a(pX)$. Given a tree t, the rule can be applied to any node with label a. Indeed, X is a free variable that matches any sequence of subtrees. If the rule is applied to a node n

Table 1. XQUF primitives, a and b are XML tags, p is a state of an HA, X, Y are free variables that denote arbitrary sequences of trees

Update Primitives	
REN	$a(X) \to b(X)$
INS_{first}	$a(X) \to a(pX)$
INS_{last}	$a(X) \to a(Xp)$
INS_{into}	$a(XY) \to a(XpY)$
INS_{before}	$a(X) \to pa(X)$
INS_{after}	$a(X) \to a(X)p$
RPL	$a(X) \to p$
DEL	$a(X) \to ()$

with label a, the result of its application is the insertion of a tree of type p as leftmost child of n. In this paper, $\Rightarrow_r$ denotes a parallel rewriting step in which the rule r is applied in a given tree t to all occurrences of subterms that match the left-hand side of r. In the previous example, we will insert a tree of type p to the left of the children of each one of the a-labelled nodes in the term t.

Target node selection is based on the node label only. In this way, indeed, we cover the whole set of XQUF primitives and still obtain an exact static analysis. More complex selection patterns could be exploited, but as [6] shows, even using basic relations (e.g. parent and ancestor), further restrictions are necessary (their method cannot support deletion in order to keep reachability decidable).

3 Schema Update Framework

In this section we propose a framework, called Schema Update Framework, that combines schema/document updates with an automata-based static analysis of their validity. Figure 2 illustrates the main components of the framework. The framework handles document adaptations expressed through the set of primitive update operations in Table 1 and schema expressed through any of the main schema definition languages for XML (DTD, XSD and RelaxNG). For each one of them, indeed, we can extract an equivalent NFHA: automata are equivalent to grammars that are in turn equivalent to schema languages [14]. More specifically, in Figure 2:

- S' is the starting schema,
- S'' is the new schema obtained from the application of a sequence of schema updates $(U_1, \ldots, U_n)$, expressed through an update language suitable to the chosen schema language,
- A is the automaton describing the new components introduced by the updates applied to schema S',
- A' is the automaton corresponding to schema S',
- A''' is the automaton recognizing the language modified using the document update sequence $(u_1, \ldots, u_m)$, where u_i is one of the primitives in Table 1, with $i \in [1 \ldots m]$,
- A'' is the automaton corresponding to the schema S''.

$$\begin{tikzcd}
S' \arrow[rr, "{(U_1,\ldots,U_m)}"] \arrow[d] & {} \arrow[d] & S'' \arrow[d] \\
A' \arrow[rr, "{(u_1,\ldots,u_n)}"'] & A & A''' \overset{?}{\subseteq} A''
\end{tikzcd}$$

Fig. 2. Summary of the framework

Note that in some cases no document adaptation may be needed (the schema update preserve document validity) or it can be derived from the schema update by a default adaptation strategy integrated by heuristics [8,2]. In the other cases, or if the default adaptation is not suitable, the user provides the sequence of document updates. For instance, in the update discussed in what follows, only the user can choose the right instance of the rule INS_{after} by fixing the type of the tree that will be inserted. As another example, if the schema update inserts a sibling of an optional element (resulting in a schema containing an optional sequence) document validity can be re-established by inserting the new element (default approach, that "mimics" the schema update) but also by deleting the original optional element.

In the initial phase, the automata A' and A'' are extracted from the schemas. The automaton A''' is computed using an algorithm that simulates the effect of the updates directly on the input HA. We then execute an *inclusion test* over the resulting HA. If the test succeeds, we can statically ensure that the application of the proposed document update sequence $(u_1, \ldots, u_m)$ on any document valid for the starting schema S' produces a document valid for the new schema S''. If the test fails, we have no guarantees that a valid document w.r.t. S', updated following the update sequence $(u_1, \ldots, u_m)$, is a valid document w.r.t. S''.

The core of the framework is the algorithm that computes the automaton A''' accepting the language obtained by the application of the document update sequence. It starts from the HA of the original schema. Depending on the operation type and parameters, it properly modifies the automaton by changing either the set of states and rules of the HA itself or the ones of the NFA accepting the horizontal languages. In this way, the semantics of the document update sequence is used to modify the language represented by the original schema, instead of the single document, allowing us to reason on the effect of the update sequence on the whole collection of documents involved. With these changes, the computed automaton accepts the original documents on which the specified sequence of updates has been applied. The algorithm and correctness proofs are available in [15]. In the following, we illustrate the behavior of the framework with the help of an example.

Listing 1.1 shows the starting schema S', expressed using XML Schema. The symbol `(*)` appearing in the schema represents the insertion point of the XML Schema fragment, referred to as S, shown in Listing 1.2. The schema update inserts the fragment S and generates the new schema S''. The three NFHA A, A' and A'', correspond to schemas S, S' and S'', respectively. For the sake of clarity, in what follows $student\text{-}info$ and $academic\text{-}transcript$ are abbreviated as si and at, respectively.

Listing 1.1. Starting schema S'.

```
<?xml version=”1.0” encoding=”utf−8”?>
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>
  <xsd:element name=”at”>
    <xsd:complexType>
      <xsd:sequence>
        (∗)
        <xsd:element name=”record” minOccurs=”0” maxOccurs=”unbounded”>
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name=”exam” type=”xsd:string”/>
              <xsd:element name=”grade” type=”xsd:string”/>
              <xsd:element name=”date” type=”xsd:date”/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

- $A = (\Sigma_L = \{'si', 'id', 'name', 'surname'\}, Q = \{p_{st}, p_i, p_n, p_s\}, Q_f = \{p_{st}\}, \Theta = \{id(\epsilon) \rightarrow p_i, name(\epsilon) \rightarrow p_n, surname(\epsilon) \rightarrow p_s, si(p_i p_n p_s) \rightarrow p_{st}\})$,
- $A' = (\Sigma = \{'at', 'record', 'exam', 'grade', 'date'\}, Q' = \{q_e, q_g, q_d, q_r, q_a\}, Q'_f = \{q_a\}, \Delta' = \{exam(\epsilon) \rightarrow q_e, grade(\epsilon) \rightarrow q_g, date(\epsilon) \rightarrow q_d, record(q_e q_g q_d) \rightarrow q_r, at(q_r^*) \rightarrow q_a\})$,
- $A'' = (\Sigma := \Sigma \cup \Sigma_L = \{'at', 'record', 'exam', 'grade', 'date', 'si', 'id', 'name', 'surname'\}, Q'' = Q' \cup Q = \{q_e, q_g, q_d, q_r, q_a, p_{st}, p_i, p_n, p_s\}, Q''_f = Q'_f = \{q_a\}, \Delta'' = \Theta \cup (\Delta' \setminus \{at(q_r^*) \rightarrow q_a\}) \cup \{at(p_{st} q_r^*) \rightarrow q_a\})$.

Listing 1.2. Schema fragment S inserted into S' by the schema update operation.

```
<xsd:element name=”si”>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name=”id” type=” xsd:string ”/>
      <xsd:element name=”name” type=”xsd:string”/>
      <xsd:element name=”surname” type=”xsd:string”/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Listing 1.3. An example of XML document D' valid w.r.t. schema S'.

```
<?xml version="1.0" encoding="utf-8"?>
<academic-transcript>
  (*)
  <record>
    <exam>Database</exam>
    <grade>25</grade>
    <date>2010-01-25</date>
  </record>
  <record>
    <exam>Calculus</exam>
    <grade>30</grade>
    <date>2010-02-25</date>
  </record>
</academic-transcript>
```

Listing 1.4. XML document fragment D.

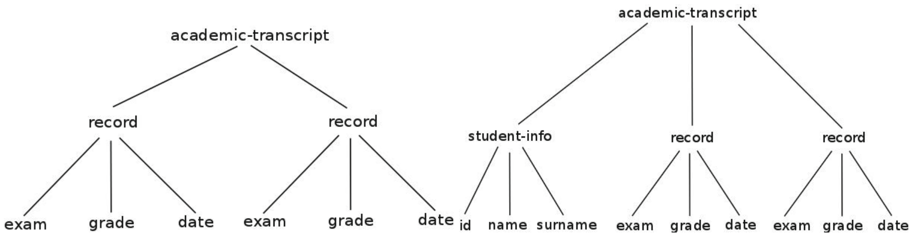
```
<si>
  <id>1234ABC</id>
  <name>Alessandro</name>
  <surname>Solimando</surname>
</si>
```

Listing 1.3 shows an example XML document, called D', valid w.r.t. schema S'. If we insert the XML schema fragment D (Listing 1.4) into D' at the insertion point marked with `(*)`, we obtain a document D'' valid w.r.t. schema S''. Figure 3(a) shows the tree t' corresponding to document D', while in Figure 3(b) we can see tree t'' corresponding to the document D''. Using operation INS_{first} we have: $at(X) \rightarrow at(tX)$, where $t = si(id, name, surname)$ that is, the tree corresponding to the XML fragment D. t'' can be easily obtained from t' ($t' \Rightarrow_{INS_{first}} t''$).

The update sequence corresponding to the suggested schema update consists in a single application of INS_{first} operation: $at(X) \rightarrow at(p_{st}X)$. The NFHA A''' accepting the language that reflects the application of the proposed document update sequence is as follows:

$A''' = (\Sigma := \Sigma \cup \Sigma_L = \{'at', 'record', 'exam', 'grade', 'date', 'si', 'id', 'name', 'surname'\}, Q' \cup Q = \{q_e, q_g, q_d, q_r, q_a, p_{st}, p_i, p_n, p_s\}, Q'_f = \{q_a\}, \Delta''')$.

Let us now see how the algorithm computes Δ'''. First of all, we need to analyze the NFA that will be modified by the algorithm, that is $B_{a-t,q_a} = (Q' \cup Q, \{b\}, b, \{b\}, \{(b, q_r, b)\})$. We then create a fresh state q^{fresh}_{a-t,q_a} and add it to the set of

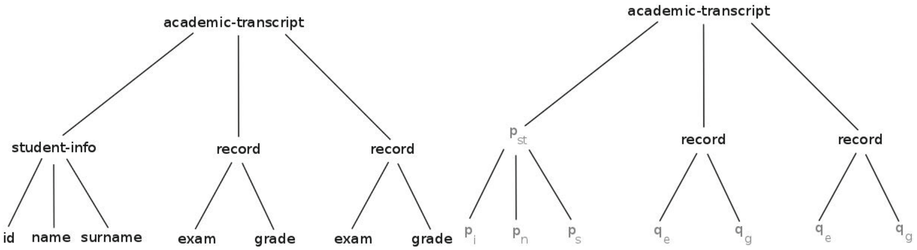


(a) Tree representation t' of document D'. (b) Tree representation t'' of document D''.

Fig. 3. Tree representations of the XML documents D' and D''

states as a starting state. We also add the rule $(q^{fresh}_{a-t,q_a}, p_{st}, b)$, since the rule (b, q_r, b) is present. After these changes, we have $B_{a-t,q_a} = (Q' \cup Q, \{b, q^{fresh}_{a-t,q_a}\}, q^{fresh}_{a-t,q_a}, \{b\}, \{(q^{fresh}_{a-t,q_a}, p_{st}, b), (b, q_r, b)\})$ and the horizontal language associated with the rule $a - t(L_{a-t,q_a}) \to q_a$ changes from q_r^* to $p_{st}q_r^*$. Finally, we compute $\Delta''' := \Theta \cup \{a(B_{a,q}) \to q \mid a \in \Sigma, q \in Q_L, L(B_{a,q}) \neq \emptyset)\}$, which is equal to $\Theta \cup (\Delta' \setminus \{a - t(q_r^*) \to q_a\}) \cup \{a - t(p_{st}q_r^*) \to q_a\}$, that is, in turn, equal to Δ''.

Because the two automata, A'' (derived from schema S'') and A''' (obtained by the algorithm that calculates the language that reflects the application of the document update sequence) are identical, the inclusion test $(A'') \subseteq L(A''')$ succeeds. The proposed update sequence is thus type safe and its application on a document valid w.r.t. S' yields a document valid w.r.t. S'', as we have already seen for the documents D' and D''.



(a) Tree representation t''' of document D'''.

(b) Computation $A''||t'''$ of the automaton A'' relative to tree t''' representing document D'''.

Fig. 4. Non accepting computation $A''||t'''$ (right) of automaton A'' over tree t''' (left)

Suppose now we modify document D'' through update $DEL : date(X) \to ()$, obtaining document D''', identical to D'' but without $date$ elements; its tree representation t''' is shown in Figure 4(a). Clearly t''' can be obtained from t'' as $t'' \Rightarrow_{DEL} t'''$. Since document D''' is not valid w.r.t. schema S'', its tree representation, t''', is not included in the language accepted by the automaton A'' corresponding to S''. In Figure 4(b) we can see the computation $A''||t'''$ of the automaton A'' related to tree t'''. This computation cannot assign an accepting state to the tree root node because the tree is not part of the language accepted by the considered automaton, since no rules of the form $record(L) \to q_L$, where $q_e q_g \in L$, exist.

4 Related Work

Most of the work on XML schema evolution has focused on determining the impact of schema updates on related document validity [8,2,7] and programs/queries [13,7]. Concerning document adaptations, the need of supporting both automatic and user-defined adaptations is advocated in [2], but no static analysis is performed on user-defined adaptations, so that run-time revalidation of adapted documents is needed.

Concerning related work on static analysis, the main formalization of schema updates is represented by [1], where the authors take into account a subset of XQUF which deals with structural conditions imposed by tags only, disregarding attributes.

Type inference for XQUF, without approximations, is not always possible. This follows from the fact that modifications that can be produced using this language can lead to non regular schemas, that cannot be captured with existing schema languages for XML. This is the reason why [1], as well as [16], computes an over-approximation of the type set that results from the updates. In our work, on the contrary, to produce an exact computation we need to cover a smaller subset of XQUF's features. In [1], indeed, XPath's axes can be used to query and select nodes, allowing to mix selectivity conditions with positional constraints with the request to satisfy a given pattern. In our work, as well as in [16] and [9], only update primitives have been considered, thus excluding complex expressions such as "for loops" and "if statements", based on the result of a query. These expressions, anyway, can be translated into a sequence of primitive operations.[1]

Macro Tree Transducers (MTT) [11] can also be applied to model XML updates as in a Monadic Second-Order logic (MSO)-based transformation language, that does not only generalize XPath, XQuery and XSLT, but can also be simulated using macro tree transducers. The composition of MTT and their property of preserving recognizability for the calculation of their inverses are exploited to perform *inverse type inference*: they pre-compute in this way the pre-image of ill-formed output and perform type checking simply testing whether the input type has some intersection with the pre-image. Their system, as ours, is exact and does not approximate the calculation, but our more specific approach, focused on a specific set of transformations, allows for a simpler (and more efficient) implementation.

5 Conclusions

In the paper we have presented an XML schema update framework relying on the use of hedge automata transformations for the static analysis of document adaptations. A Java prototype, based on the LETHAL Library, of the framework have been realized and tested on the XML XMark benchmark. The execution times of the automaton computation and of the inclusion test on the considered bechmarks are negligible (less than 1s), showing the potential of our proposal for a practical usage as a support for static analysis of XML updates. We plan to integrate this prototype of the framework with *EXup* ([2]) or other suitable tools as future work.

Other possible directions for extending the current work can be devised. Node selection constraints for update operations could be refined, for example using XPath axes and the other features offered by XQUF. As discussed in the paper, this would lead to approximate rather than exact analysis techniques. Support for commutative trees, in which the order of the children of a node is irrelevant, could be added. This feature would allow the formalization of the *all* and *interleave* constructs of XML Schema [18] and Relax NG [3], respectively. Sheaves Automata, introduced in [5], are able to recognize commutative trees and have an expressivity strictly greater than the HA considered in this work. The applicability of these automata in our framework needs to be investigated.

[1] The interested reader could refer to [1] (Section "Semantics"), where a translation of XQUF update expressions into a pending update list, made only of primitive operations, is provided, according to the W3C specification [17].

References

1. Benedikt, M., Cheney, J.: Semantics, Types and Effects for XML Updates. In: Gardner, P., Geerts, F. (eds.) DBPL 2009. LNCS, vol. 5708, pp. 1–17. Springer, Heidelberg (2009)
2. Cavalieri, F., Guerrini, G., Mesiti, M.: Updating XML Schemas and Associated Documents through Exup. In: Proc. of the 27th International Conference on Data Engineering, pp. 1320–1323 (2011)
3. Clark, J., Murata, M.: RELAX NG Specification (2001), http://www.relaxng.org/spec-20011203.html
4. Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree Automata Techniques and Applications (2007), http://www.grappa.univ-lille3.fr/tata (release October 12, 2007)
5. Dal-Zilio, S., Lugiez, D.: XML Schema, Tree Logic and Sheaves Automata. In: Nieuwenhuis, R. (ed.) RTA 2003. LNCS, vol. 2706, pp. 246–263. Springer, Heidelberg (2003)
6. Genest, B., Muscholl, A., Serre, O., Zeitoun, M.: Tree Pattern Rewriting Systems. In: Cha, S(S.), Choi, J.-Y., Kim, M., Lee, I., Viswanathan, M. (eds.) ATVA 2008. LNCS, vol. 5311, pp. 332–346. Springer, Heidelberg (2008)
7. Geneves, P., Layaiada, N., Quint, V.: Impact of XML Schema Evolution. ACM Trans. Internet Technol. 11(1) (2011)
8. Guerrini, G., Mesiti, M., Sorrenti, M.: XML Schema Evolution: Incremental Validation and Efficient Document Adaptation. In: 5th International XML Database Symposium on Database and XML Technologies, pp. 92–106 (2007)
9. Jacquemard, F., Rusinowitch, M.: Formal Verification of XML Updates and Access Control Policies (May 2010)
10. Liu, Z., Natarajan, S., He, B., Hsiao, H., Chen, Y.: Cods: Evolving data efficiently and scalably in column oriented databases. PVLDB 3(2), 1521–1524 (2010)
11. Maneth, S., Berlea, A., Perst, T., Seidl, H.: XML type checking with macro tree transducers. In: PODS, pp. 283–294 (2005)
12. Moon, H., Curino, C., Deutsch, A., Hou, C., Zaniolo, C.: Managing and querying transaction-time databases under schema evolution. PVLDB 1(1), 882–895 (2008)
13. Moro, M., Malaika, S., Lim, L.: Preserving xml queries during schema evolution. In: WWW, pp. 1341–1342 (2007)
14. Murata, M., Lee, D., Mani, M., Kawaguchi, K.: Taxonomy of XML schema languages using formal language theory. ACM Trans. Internet Technol. 5(4), 660–704 (2005)
15. Solimando, A., Delzanno, G., Guerrini, G.: Static Analysis of XML Document Adaptations through Hedge Automata. Technical Report DISI-TR-11-08 (2011)
16. Touili, T.: Computing Transitive Closures of Hedge Transformations. In: Proc. 1st Int. Workshop on Verification and Evaluation of Computer and Communication Systems (VECOS 2007). eWIC Series. British Computer Society (2007)
17. Chamberlin, D., Dyck, M., Florescu, D., Melton, J., Robie, J., Simon, J.: W3C. XQuery Update Facility 1.0 (2009), http://www.w3.org/TR/2009/CR-xquery-update-10-20090609/
18. Walmsley, P., Fallside, D.C.: W3C. XML Schema Part 0: Primer Second Edition (2004), http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/

Supporting Database Provenance under Schema Evolution

Shi Gao and Carlo Zaniolo

University of California, Los Angeles
{gaoshi,zaniolo}@cs.ucla.edu

Abstract. Database schema upgrades are common in modern information systems, where the provenance of the schema is of much interest, and actually required to explain the provenance of contents generated by the database conversion that is part of such upgrades. Thus, an integrated management for data and metadata is needed, and the Archived Metadata and Provenance Manager (AM&PM) system is the first to address this requirement by building on recent advances in schema mappings and database upgrade automation. Therefore AM&PM (i) extends the Information Schema with the capability of archiving the provenance of the schema and other metadata, (ii) provides a timestamp based representation for the provenance of the actual data, and (iii) supports powerful queries on the provenance of the data and on the history of the metadata. In this paper, we present the design and main features of AM&PM, and the results of various experiments to evaluate its performance.

1 Introduction

The importance of recording the *provenance*, or *lineage*, of any information of significance is now widely recognized, and a large body of research was produced on provenance management for scientific workflows and databases [4,15,18]. Previously proposed provenance systems focus on capturing the "why", "where", and "how" facets of provenances [5, 13], and support a rich set of provenance-related functions and queries. Unfortunately, current proposals assume that the whole database content was created by the external actions of users and applications. In reality, modern information systems, particularly big science projects, undergo frequent schema changes whereby the database under the old schema is migrated into the new one conforming to the new schema. Thus the current database is the combined result of (i) the external actions that entered the original information (e.g., via SQL inserts or updates), and (ii) the migration steps that then transformed the data as part of the schema evolution process. Therefore explaining the provenance of current data often requires 'flashing-back' to the original schema, external transaction, and data as this was originally created, i.e., before the conversion step (or the succession of steps) that transformed the original information into the equivalent one that is now stored under the current schema. Thus supporting provenance under schema evolution can be challenging and is hardly surprising that previous works have ignored this challenge by

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 67–77, 2012.

assuming that the database schema is fixed and does not change with time. However this assumption is utterly unrealistic as illustrated by the UCLA testbed collecting the schema history for 20+ large information systems, including Mediawiki/Wikipedia, Ensembl GeneticDB and various CERN Physics DBs [8,9]. For instance, the database of Mediawiki software supporting Wikipedia has experienced more than 300 schema versions in its nine years of life and similar observations hold for the other information systems. Fortunately, recent results obtained by the UCLA Panta Rhei project [10,11] have significantly simplified the problems of preserving the history of the database schema and flashing back to the original schema and source data.

The results presented in [10,11] provide the conceptual basis for the Archived Metadata and Provenance Manager (AM&PM) described in this paper; AM&PM is the first system designed to support the provenance of both data and metadata in the presence of schema evolution. Thus, AM&PM (i) extends the SQL information schema with timestamp based archival, (ii) uses Schema Modification Operators (SMOs) and Integrity Constraints Modification Operators (ICMOs) [10,11] to map between different versions of the database schema, and (iii) supports powerful queries for tracing the provenance of data and metadata, to support functions such as database auditing and data-recovery [6,18].

The reminder of this paper is organized as follows: In Section 2, we review the schema evolution language used to describe schema evolution. In Section 3, we present the details of our approach for archiving the provenance of data and metadata. In Section 4, we discuss the design and features of the AM&PM system. Section 5 presents an evaluation on the space and time required by the provenance tables and queries. Related work is summarized in Section 6, and our conclusions are reported in Section 7.

2 Schema Evolution Languages

The Schema Modification Operators (SMOs) were introduced in [10] as a very effective tool for characterizing schema upgrades and automating the process of migrating the database and upgrading the query-based applications. Since in many situations, the schema integrity constraints are also changed along with schema structure, Integrity Constraints Modification Operators (ICMOs) were introduced in [11] to characterize integrity constraints and automate the process of upgrading applications involving integrity constraint updates. Three types of integrity constraints are considered: primary key, foreign key, and value constraint. Extensive experience with the schema evolution testbed [8] shows that the combination of SMOs and ICMOs displayed in Table 1 can effectively capture and characterize the schema evolution history of information systems. In Table 1, (R, S) denote relational tables and (a, b, c, d) denote columns in tables. Here we present a simple example to show how schema evolution language works.

Table 1. Schema Evolution Language: SMO and ICMO

SMO
CREATE TABLE R(a,b,c)
DROP TABLE R
RENAME TABLE R INTO T
COPY TABLE R INTO T
MERGE TABLE R, S INTO T
PARTITION TABLE R INTO S WITH cond, T
DECOMPOSE TABLE R INTO S(a,b), T(a,c)
JOIN TABLE R,S INTO T WHERE conditions
ADD COLUMN d INTO R
DROP COLUMN c FROM R
RENAME COLUMN b IN R TO d

ICMO
ALTER TABLE R ADD PRIMARY KEY pk1(a; b) [policy]
ALTER TABLE R ADD FOREIGN KEY fk1(c; d) REFERENCES T(a; b) [policy]
ALTER TABLE R ADD VALUE CONSTRAINT vc1 AS R:e = 0 [policy]
ALTER TABLE R DROP PRIMARY KEY pk1
ALTER TABLE R DROP FOREIGN KEY fk1
ALTER TABLE R DROP VALUE CONSTRAINT vc1

Example 1: Consider one database upgrade scenario as follows:

V_1: *Employee(ID, Name, Pay)*
V_2: *Employee_Info(ID, Name)* *Employee_Pay(ID, Salary)*

The initial schema version is V_1. Then the schema evolves into V_2 and the table *Employee* is decomposed into two tables: *Employee_Info* and *Employee_Pay*. Underlines indicate the primary keys. The transformation from V_1 to V_2 can be described as:

1. RENAME COLUMN *Pay* IN *Employee* To *Salary*
2. DECOMPOSE TABLE *Employee* INTO *Employee_Info(ID, Name)*, *Employee_Pay (ID, Salary)*

The composition of these two SMOs produces a *schema evolution script*, which defines the mapping between different versions of database schema in a concise and unambiguous way. Schema evolution scripts can be characterized as Disjunctive Embedded Dependencies (DEDs) and converted into SQL scripts [10]. We will use the schema evolution scripts to archive schema changes and perform provenance query rewriting, as discussed in Section 4.

3 Provenance Management in AM&PM

The AM&PM system is designed to provide the integrated management for the provenance of data and metadata, in order to support a range of queries, including the following ones:

Data Provenance Queries: Given a data tuple d in the current database, where does d come from (e.g. schema, table, and column)? When was d created? and how (e.g. by which transaction and user)?

Schema Provenance Queries: AM&PM allows users to retrieve past versions of database schema along with the SMOs and ICMOs which describe the schema evolution process in databases. Similarly, we can answer where-, when-, and how-queries for the basic schema elements in databases.

Evolution Statistic Queries: AM&PM supports statistical queries about the evolution of the database content and schema.

Information Schema History. In a relational database management system, the *information schema*, also called *data dictionary* or *system catalog*, is the database for the metadata. As specified in the SQL:2003 standards [12], the information schema consists of a set of tables and views that contain information about the schema and other metadata. In order to answer queries on past versions of the schema and the provenance of the current database, AM&PM stores the transaction time history of the information schema.

Data Provenance. The provenance of each attribute value in the current database is recorded by preserving the information about the transactions that generated it, including the transaction ID, the timestamp at which the transaction executed, and the user who issued the transaction. Thus, AM&PM introduces the tables *Transaction* and *Tran_Text* to archive the transactions with their timestamps. The mapping tables are introduced to specify the relationship between transactions and tuples. This is all it is needed for data generated under the current versions of the schema. However, the current data could be the result of database migration after schema upgrades. Therefore, we must use the SMOs and ICMOs to reconstruct the source schema and data value, using the process described next under ‘schema provenance’.

Schema Provenance. Schema provenance refers to the combination of the past versions of database schema and the schema evolution history. AM&PM exploits the information schema tables to store the metadata of past schemas. Two tables, *SMO* and *ICMO*, are created to archive the schema evolution history. Every schema change in the database upgrade is converted to a schema evolution operator (SMO or ICMO) and stored with its timestamp. Examples of these two tables are shown in tables (e) and (f) of Figure 1. The tuple *s1* in the *SMO* table represents one schema change in the transformation from V_1 to V_2.

Auxiliary Information. AM&PM can also support some optional tables to provide more information about provenance (e.g., the values before the last data update and statistics).

Example 2: Figure 1 shows the provenance database for the case of *Example 1.* The relationship between data and transactions is stored in mapping tables (c) and (d). For instance, the transaction which affects the value “Sam” in attribute *Name* of table *Employee_Info* can be found in table *Employee_Info_Mapping*, as “t1”.

(a) Employee_Info

ID	Name
100	Sam
200	Bob

(b) Employee_Pay

ID	Salary
100	2000
200	3000

(c) Employee_Info_Mapping

Tuple_ID	Attribute	TID
100	*ID*	t1
100	*Name*	t1
200	*ID*	t2
200	*Name*	t2

(d) Employee_Pay_Mapping

Tuple_ID	Attribute	TID
100	*ID*	t1
100	*Salary*	t1
200	*ID*	t2
200	*Salary*	t2

(e) SMO

SID	SMO	Source	Target	Timestamp
s1	smo_1	V_1	V_2	2007-11-20 12:38:44
s2	smo_2	V_1	V_2	2007-11-20 12:42:07

(f) ICMO

ICID	ICMO	Source	Target	Timestamp
i1	$icmo_1$	V_2	V_3	2008-02-01 13:04:46

(g) Transaction

TID	DB_User	Schema	Timestamp
t1	Guest1	V_1	2006-09-15 11:09:12
t2	Guest1	V_1	2006-10-16 17:26:09

(h) Tran_Text

TID	Trans
t1	$tran_1$
t2	$tran_2$

(i)

smo_1: RENAME COLUMN *Pay* IN *Employee* To *Salary*
smo_2: DECOMPOSE TABLE *Employee* INTO *Employee_Info(ID, Name)*, *Employee_Pay(ID, Salary)*
$icmo_1$: ALTER TABLE *Employee_Info* ADD FOREIGN KEY pk1 (*ID*) REFERENCE *Employee_Insurance* (*Employee_ID*)
$tran_1$: INSERT INTO *Employee* VALUES (100, ‘Sam’, 2000)
$tran_2$: INSERT INTO *Employee* VALUES (200, ‘Bob’, 3000)

Fig. 1. An example of AM&PM provenance database

This transaction id denotes $tran_1$ as the transaction that created/updated the value. The provenance queries are answered by evaluating the timestamps. For instance, say that a user issues a how-provenance query of the value “Sam” to verify this data. The last transaction which affects this data is $tran_1$ with timestamp “2006-09-15 11:09:12”. All the schema evolution operators that have occurred from this timestamp until now may affect the data. Thus, we obtain smo_1, smo_2, and $icmo_1$, but then we find that only smo_1 and smo_2 affect the attribute *Name*. We return the timestamped $tran_1$, smo_1, and smo_2 as the how-provenance of “Sam”.

4 AM&PM System

Figure 2 depicts the high-level architecture of AM&PM system. The AM&PM system consists of two main components: *the provenance database manager* and *the query parser.*

The AM&PM provenance database manager analyzes the input data and constructs the provenance database, which is a combined relational database

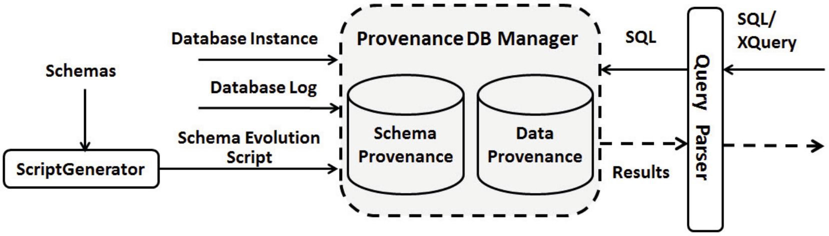


Fig. 2. Architecture of AM&PM system

of *schema provenance tables* and *data provenance tables*. Schema provenance tables include the schema information of past database versions, SMOs, ICMOs, and database statistics, while data provenance tables (e.g. transaction table) store the information of transactions applied to the content of database and the relationship between transactions and data. The schema provenance tables are filled by parsing the past schemas and the schema evolution scripts. A schema evolution script is the set of SMOs and ICMOs used to upgrade database, as defined in Section 2. In AM&PM, the schema evolution script is produced by a module called *ScriptGenerator*. The basic idea of the ScriptGenerator is to compare different schemas and express the changes using the schema evolution language. The content of data provenance tables is extracted from the current database and the database transaction log. The database transaction log offers the source values and relevant transactions.

Once the provenance database is filled, users can issue the provenance queries in SQL or XQuery. Many provenance queries are temporal queries with time constraints, and XQuery can facilitate the expression of complex temporal provenance queries [19]. Provenance queries written in SQL are directly executed in our provenance database. However, queries expressed in XQuery are rewritten to SQL statements by the AM&PM query parser. For that, we utilize the algorithm presented in [19], based on the SQL/XML standard [16].

Extended Functionality: At the present time, AM&PM supports the queries and functions described above; we plan to support the functions described next in future extensions.

Provenance Query Rewriting Suppose we have a provenance query set, and some changes occur in the current database schema. Can the current provenance query be automatically upgraded to work on the new schema? To solve this problem, we are investigating the extensions of query rewriting techniques proposed in PRISM++ [11].

Provenance Propagation in Data Warehouses If a value in the source database is changed, how to propagate this change to the current database? This problem is studied in [14] in data warehousing and we plan to develop similar techniques in AM&PM.

5 Experimental Evaluation

Our experiments seek to evaluate (i) the space overhead of provenance capture, and (ii) the execution time of provenance queries over AM&PM provenance database. Note that there is no comparison with other provenance management systems since AM&PM is the first system using a temporal model to archive the provenance of data and metadata under schema evolution. Most queries discussed in Section 3 are not supported by other existing systems.

Experiment Setup. Our experiments used four real-world data sets: Mediawiki, Dekiwiki, KtDMS, and California Traffic [1,8]. The first three data sets are the database schemas of web information systems. As shown in Table 2, these three information systems experienced many schema changes during their lifetime. We obtained the database schemas of these information systems from [8] and used them in our experiments on schema provenance.

In AM&PM, the database transaction log is required to fill the data provenance tables. Thus, we constructed a California Traffic database according to the website of California transportation department [1]. Every update of highway information in the website was captured and rewritten to a transaction applied to traffic database.Thus, there are two main tables in the traffic database: *highway_accident* and *highway_condition*. For instance, when there is a new accident in website, a transaction inserting a tuple to *highway_accident* is generated in the traffic database. Our real-world California Traffic database contains 5369 transactions. Since the California Traffic database is relatively small, we also generated four large synthetic traffic databases to verify the scalability of our approach.

The values in synthetic traffic databases are uniformly sampled from the real-world California Traffic database. The number of transactions and the size of data provenance tables of the four synthetic traffic databases are (50000, 22M), (100000, 42M), (200000, 91M), (500000, 231M) respectively.

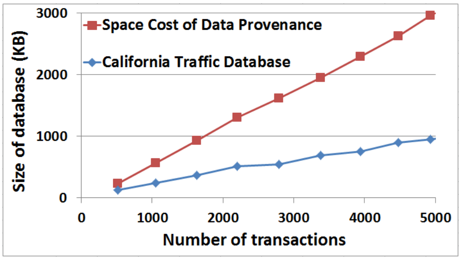


Fig. 3. Size of data provenance tables

Table 2. Size of schema evolution tables

System	Lifetime (years)	# of Versions	Space (MB)
Dekiwiki	4.0	16	0.933
KtDMS	6.3	105	10.717
Mediawiki	9.0	323	18.331

Our experiment environment consisted of a workstation with Intel Xeon E5520 CPU and 4GB RAM running Ubuntu 11.10. MySQL 5.1.61 was used as the backend database for AM&PM. We restarted the database server between query runs and used the average of five runs to report our the results.

Space Overhead of Provenance Capture: The AM&PM provenance database consists of data provenance tables and schema provenance tables, as shown in Section 4. We evaluate the space overhead of these two parts respectively.

Data Provenance Tables Figure 3 shows the space overhead of such tables as the number of transactions in the real-world California Traffic database increases. We observe that the size of data provenance tables is proportional to the number of transactions. Since AM&PM allows users to retrieval the

Table 3. Data provenance queries for evaluation

Query	Description
Q1	find the creation time of the tuple with id 2357 in highway_accident
Q2	find the transaction which generates the tuple with id 19009 in highway_condition
Q3	find the number of accidents happening on 04/02/2012
Q4	find the number of highway condition records on 04/03/2012
Q5	find the ids of accidents happening in the area of West Los Angeles between '04/04/2012 18:00:00' and '04/04/2012 23:00:00'
Q6	find the descriptions of highway condition updates happening in the area of Central LA between '04/04/2012 18:00:00' and '04/04/2012 23:00:00'

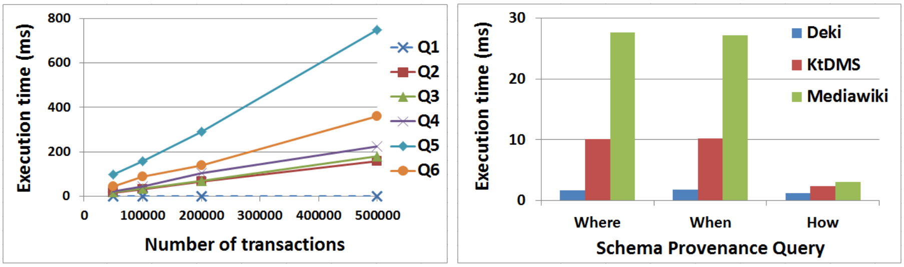
content of transactions, the space overhead for storing transactions is unavoidable. The relationship between the space overhead of data provenance tables and the size of California Traffic database is also approximately linear. Archiving the data provenance leads to about 300% space overhead for the traffic database. This high relative overhead reflects the fact that the transactions and timestamps require several bytes, whereas the data elements in the traffic database are small—e.g., names of highways and regions.

Schema Provenance Tables. We investigate the space overhead of schema provenance tables in three information systems: Mediawiki, Dekiwiki, and KtDMS, as shown in Table 2. The versions in Table 2 are defined according to their SVN (Apache Subversion) versions. The result shows that the space cost of schema provenance tables depends on the number of history schema versions. KtDMS and Mediawiki need more space for schema provenance tables than Dekiwiki because they have more database versions and schema changes. In particular, for Mediawiki which experienced many schema changes, it only takes 18 MB to store the schema evolution history of 323 schemas. Comparing with the size of Wikipedia database [2] under the latest schema, which is about 34.8 GB uncompressed, the space overhead of schema provenance tables is very small. Therefore, the space overhead of schema provenance is typically modest.

Query Execution Time: We study the query performance by measuring the execution time. To be specific, the execution time refers to the time used by the backend database to execute queries written in standard SQL. The types of provenance queries tested are those discussed in Section 3.

Data Provenance Queries. Since the real-world California Traffic database is relatively small, most provenance queries are answered within 1 ms. We use the synthetic traffic databases to evaluate the performance of data provenance queries. Table 3 shows the set of data provenance queries we prepared for experiments. Q1 and Q2 are the when- and how- provenance queries. Q2 also involves join operation. Q3 and Q4 are temporal provenance queries with aggregates. Q5 and Q6 are temporal provenance queries with joins.

Figure 4a shows the query execution time as the number of transactions in the synthetic traffic databases increases. The performance of simple provenance



(a) Data provenance query

(b) Schema provenance query

Fig. 4. Execution time for data and schema provenance queries

query like Q1 is not affected by the number of transactions. For all of the four synthetic databases, it takes almost the same time (around 0.4 ms) to execute Q1. The execution time of other provenance queries is linear with the number of transactions. For queries with aggregates, they have to scan the *transaction* table to get the count. For queries with joins, they require to join the data provenance tables whose size is proportional to the number of transactions as shown in Figure 3.

Schema Provenance Query. We evaluate the performance of schema provenance query using Dekiwiki, KtDMS, and Mediawiki. For each information system, we randomly pick 5 tables and 5 columns. Then the where-, why-, and how-provenance queries of these tables and columns are executed as test queries. The results of where-, when-, and how- queries on schema provenance are as follows: the source schema, the creation time and the schema evolution operators which help generate the tables and columns. Figure 4b shows the average execution time of schema provenance queries. The result indicates that the performance of schema provenance query depends on the size of schema evolution tables. It takes more time to run schema provenance queries in the system which has larger schema provenance tables. We also observe that the how-query takes less time because it only scans the SMO or the ICMO table, while where- and when-queries need to join the information schema tables (e.g. COLUMNS and VERSIONS) to associate the schema element with its version and timestamp.

Statistic Query. We prepared a set of statistic queries to compute five statistics for each version of database schema: the number of tables, the number of columns, the number of primary keys, the number of foreign keys, and the lifetime. The overall execution time of statistic query set is 16.6 ms for Deikiwiki, 131.0 ms for KtDMS, and 217.3 ms for Mediawiki. Not surprisingly, the execution time of these statistic queries is linear with the size of schema provenance tables.

6 Related Work

There has been a large body of research on the provenance in databases and, because of space restrictions, we will refer our users to the overview papers [4,6,15].

The differences between where- and why- provenance in databases were studied in [5], while Green et al. [13] studied the how-provenance. The challenge of archiving the provenance of metadata was discussed in [18]. Schema mapPIng DEbuggeR (SPIDER) [3,7] used nested relational model and schema mapping language to record the data mapping between two schemas. Provenance is computed according to the routes of possibly mappings. But [3,7] didn't propose a model to archive the schema changes in schema evolution. On the other hand, the Metadata Management System (MMS) [17] comes closer to our proposal. To associate data and metadata, MMS stores the queries as values in a relational database. Traditional join mechanisms are extended to support the join specified by the queries stored in relations. However, the evolution of metadata is not considered in MMS.

7 Conclusion and Future Work

In this paper, we present an integrated approach to manage the provenance of both data and metadata under schema evolution. The provenance of data and metadata is archived using relational databases. The schema evolution history is stored using schema evolution language. Thus, AM&PM provides a simple way to support provenance management in relational databases. Powerful provenance queries expressed in SQL and XQuery are efficiently supported. We perform experiments on several real-world and synthetic data sets. The results validate the practicality of our approach.

The work presented in this paper is the first system for the provenance management under schema evolution, and leaves many topics open for further studies and improvements which were briefly discussed in the paper.

Acknowledgement. The authors would like to thank Mohan Yang, Kai Zeng, and Seyed Hamid Mousavi for their insightful comments and suggestions on this research. This work was supported in part by NSF Grant No. IIS 0917333, entitled "III: Small: Information Systems Under Schema Evolution: Analyzing Change Histories and Management Tools."

References

1. California department of transofrmation-highway conditions, http://www.dot.ca.gov/hq/roadinfo
2. Wikipedia:database download, http://en.wikipedia.org/wiki/Wikipedia:Database_download

3. Alexe, B., Chiticariu, L., Tan, W.C.: Spider: a schema mapping debugger. In: VLDB, pp. 1179–1182 (2006)
4. Bose, R., Frew, J.: Lineage retrieval for scientific data processing: a survey. ACM Comput. Surv. 37(1), 1–28 (2005)
5. Buneman, P., Khanna, S., Tan, W.-C.: Why and Where: A Characterization of Data Provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 316–330. Springer, Heidelberg (2000)
6. Buneman, P., Tan, W.C.: Provenance in databases. In: SIGMOD Conference, pp. 1171–1173 (2007)
7. Chiticariu, L., Tan, W.C.: Debugging schema mappings with routes. In: VLDB, pp. 79–90 (2006)
8. Curino, C., Zaniolo, C.: Pantha rhei benchmark datasets, http://yellowstone.cs.ucla.edu/schema-evolution/index.php/Benchmark_home
9. Curino, C., Moon, H.J., Tanca, L., Zaniolo, C.: Schema evolution in wikipedia - toward a web information system benchmark. In: ICEIS, pp. 323–332 (2008)
10. Curino, C.A., Moon, H.J., Zaniolo, C.: Graceful database schema evolution: the prism workbench. Proc. VLDB Endow. 1(1), 761–772 (2008)
11. Curino, C.A., Moon, H.J., Deutsch, A., Zaniolo, C.: Update rewriting and integrity constraint maintenance in a schema evolution support system: Prism++. Proc. VLDB Endow. 4(2), 117–128 (2010)
12. Eisenberg, A., Melton, J., Kulkarni, K.G., Michels, J.-E., Zemke, F.: Sql: 2003 has been published. SIGMOD Record 33(1), 119–126 (2004)
13. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: PODS, pp. 31–40 (2007)
14. Ikeda, R., Salihoglu, S., Widom, J.: Provenance-based refresh in data-oriented workflows. In: CIKM, pp. 1659–1668 (2011)
15. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. 34(3), 31–36 (2005)
16. SQL/XML, http://www.sqlx.org/
17. Srivastava, D., Velegrakis, Y.: Intensional associations between data and metadata. In: SIGMOD Conference, pp. 401–412 (2007)
18. Tan, W.C.: Provenance in databases: Past, current, and future. IEEE Data Eng. Bull. 30(4), 3–12 (2007)
19. Wang, F., Zaniolo, C., Zhou, X.: Archis: an xml-based approach to transaction-time temporal database systems. The VLDB Journal 17(6), 1445–1463 (2008)

An in-Browser Microblog Ranking Engine

Stéphane Frénot[1] and Stéphane Grumbach[2]

[1] Université de Lyon
[2] INRIA

Abstract. Microblogs, although extremely peculiar pieces of data, constitute a very rich source of information, which has been widely exploited recently, thanks to the liberal access Twitter offers through its API. Nevertheless, computing relevant answers to general queries is still a very challenging task. We propose a new engine, the Twittering Machine, which evaluates SQL like queries on streams of tweets, using ranking techniques computed at query time. Our algorithm is real time, it produces streams of results which are refined progressively, adaptive, the queries continuously adapt to new trends, invasive, it interacts with Twitter by suggesting relevant users to follow, and query results to publish as tweets. Moreover it works in a decentralized environment, directly in the browser on the client side, making it easy to use, and server independent.

1 Introduction

The amounts of personal data accumulated at Internet scale constitute a resource, much like raw materials, with a huge economic potential. Google made the first demonstration of the power of these data in 2003 with its Flu Trend, which allows to monitor the flu activity in the world, based on the frequency in all languages of flu related search on its engine, and has been shown not only to be accurate but moreover ahead of disease control institutions [GMP$^+$09].

Among the large Internet corporations handling users data, Twitter is the one that provides the most liberal access to them. Since its inception in 2006, Twitter grew at an exponential pace[1], with now over 100 million active users, producing about 250 million tweets daily. Although microblogs might seem restricted, Twitter has an amazing potential, it serves more than a billion queries per day, supports around a million Apps[2], and its projected advertising revenue of $250 million in 2012 is predicted to have doubled by 2014.

In this paper, we propose a new approach to handle social network data. Our objective is twofold. (i) First, develop **real time algorithms** to find the **most relevant data on any topic**, whether popular or not, by using any ranking techniques. (ii) Second, develop **continuous server-less solutions**, with algorithms running on the client side.

[1] http://www.dazeinfo.com/2012/02/27/twitter-statistics-facts-and-figures-from-2006-2012-infographic/

[2] http://gigaom.com/2011/08/16/twitter-by-the-numbers-infographic/

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 78–88, 2012.

On Twitter, our first objective is to identify the most relevant tweets satisfying a query, the most relevant twitterers on a topic or the most important tags to listen to. Our notion of relevance is similar to a topic specific Pagerank, which could rely on any concept available in the social network system.

Extracting knowledge from tweets is a challenging task. Microblog data are rather unconventional. On one hand, they are like most web data, relying on both content and graph connections (followers, links to URL, etc.), but on the other hand, their content is rather peculiar, both for its form, extremely short, and for its substance, often very communication or notification oriented. Moreover, inputs as well as most generally outputs, are streams of data, which should be handled as most as possible in real time.

Our algorithms have the following characteristics. (i) The rankings are computed at query time, streams of results are produced in realtime, with an accuracy which increases progressively. (ii) Queries are autonomic since they are continuously refined by taking into accounts trends expressed in the output streams. (iii) The algorithms are associated with a twitterer, and interacts with Twitter, by publishing query results, and following relevant twitterers.

Our second objective is to develop systems which are server-less and work as soon as the client's side is active. This approach is very promising. First, functionalities which are not offered by Internet corporations, can be handled directly by the users on their CPU. More generally, this model of decentralized computation alleviates the burden of centralized systems, by pushing to the periphery computations that do not need to be centralized, as Facebook does with the Like button for instance. Complex computations on fragments of large data sets can be performed as well at the client side, as is done by systems such as seti@home.

We demonstrate our approach on Twitter, with the Twittering Machine, that we developed in javascript, and which runs directly in the browser of the client. It is associated with the personal account of the user, which will be used to follow relevant twitterers as well as publish query results, and works essentially as follows. The machine takes as input streams of tweets satisfying an initial query, which are immediately displayed and analysed. The most relevant twitterers are then computed, and it is suggested to follow them, adding to the input of the machine their tweets. The keywords in the query are modified according to query results. In fact, the query takes a stream of keywords as parameter. Query results are regularly suggested for publication.

The paper is organized as follows. In the next section, we present the Twittering Machine. Section 3 is devoted to the presentation of the plug-ins specifying the queries, while some experimental results are presented in Section 4.

2 The Twittering Machine

Twitter offers essentially two ways to access tweets, (i) by following specific already known twitterers, with their complete stream of tweets, or (ii) by using the search API. The latter provides a very efficient real time search [BGL$^+$12], with a simple query language that allows to filter tweets on their content, as well as on various attributes, such as their hashtags, URLs, locations, etc. It produces a small percentage of the theoretical result, but often sufficient to extract knowledge, and is widely used.

The main goal of the Twittering Machine is to evaluate complex queries over streams of tweets. It relies on computations local to the client side, and works autonomously, without any server except the queried social system, Twitter. This section presents its global functionality, illustrated in Fig. 1, and explains the implementation choices.

The machine takes as inputs incoming streams of tweets either results of search queries, or produced by followed twitterers, performs algebraic computation corresponding to stream queries, and produces as output, streams of tweets to display to the user, tweets to be produced by the user on a specific topic, as well as instructions for the management of followed twitterers. Output results are re-injected into the main system, thus leading to an autonomic querying system, which self-regulates.

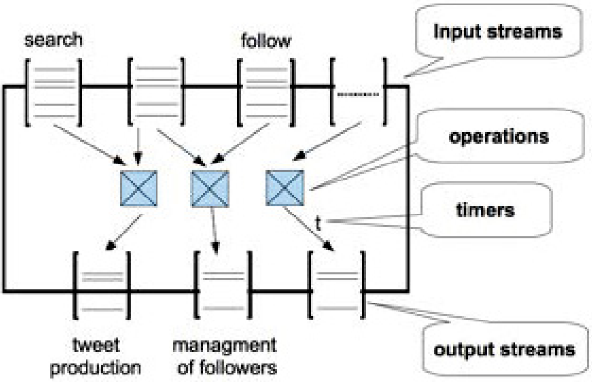


Fig. 1. The Twittering Machine

The Twittering Machine has the following main characteristics.

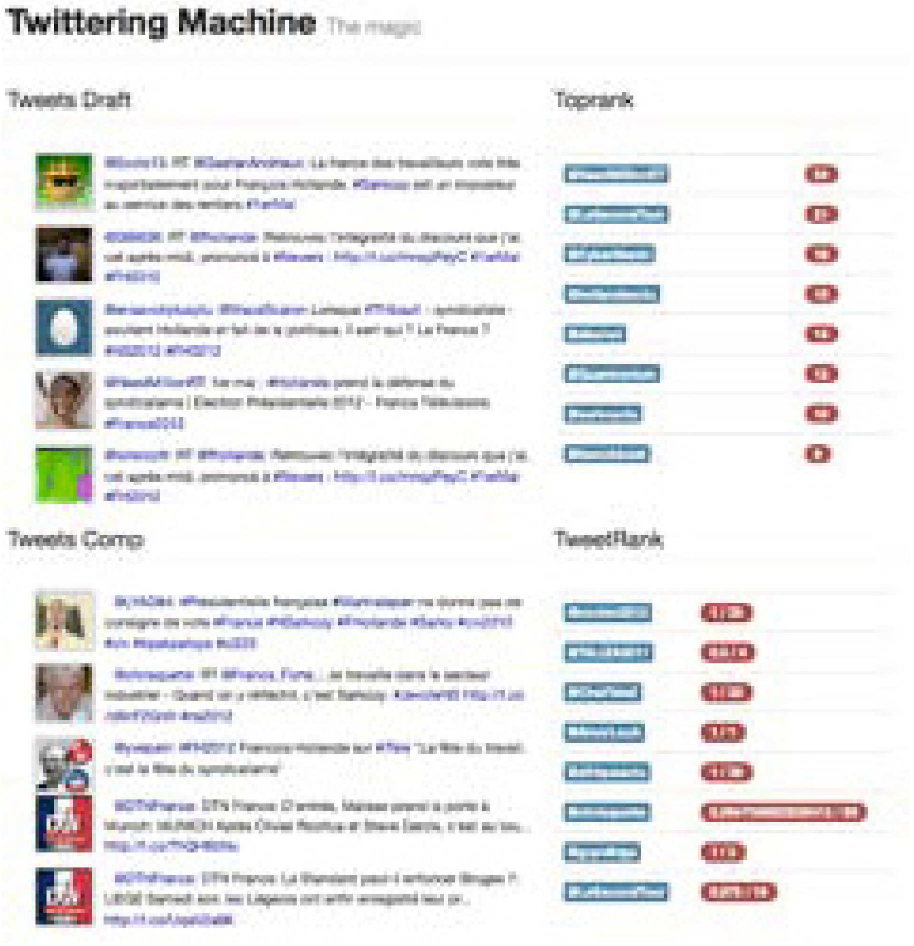
In-node execution We adopt a decentralized approach, with a system running mainly at the border of the network, on the end-user's computer, and to simplify its use, we choose to run directly at the browser level. With the rapid development of cloud architectures, more and more run-times are hidden at the clients side. Whenever possible, we always provide in-browser function execution. For cross-domain browser limitation reasons[3], we use a local run-time based on nodejs[4] that communicates locally with the browser and handles authentication requests. From the Twitter side, our in-node approach behaves as a standard end-user and develops a behavior similar to a real user.

Twitter friendliness The Twittering Machine interacts naturally with the API of Twitter. It uses all functionalities Twitter offers in its API. Twitter's data share

[3] http://en.wikipedia.org/wiki/Same_origin_policy

[4] http://nodejs.org/

with other web data, the duality of graph connections (followers) and contents (tweets), but its content is unconventional both in form and in substance. Tweets can be grabbed either from the general search API or from the followed twitterers through either REST or the stream based authenticated API. Followers can be managed with the same authenticated API as the REST API.



Twittering Machine The magic

Tweets Draft

Toprank

Tweets Comp

TweetRank

Fig. 2. The displayed tweets with their counts

Stream based plugin architecture The Twittering Machine is build around dynamic modules orchestrated into the browser. Each plugin works in a stream based way. It manages an input stream generally of tweets obtained through a dedicated query, realizes a grouping and an ordering of tweets and finally generates a stream for instance of twitterers as output. Plugins can be cascaded since the output from one plugin may be used as the input of another, as shown in Section 3. Finally, we consider every input and output parameters as streams. For instance the query parameters are extracted from an input stream and this input stream can be adapted at run-time to include relevant keywords for instance that pop-up from the output computed so far.

Specifying plugins with a query Language Every plugin is expressed with a stream based query that uses the select-project-join-aggregate syntax of SQL as presented in Section 3. The ranking is defined using such queries which recursively lead to more accurate results.

Interaction with Twitter The Twittering Machine can suggest interactions with Twitter to the user, such as manage its follow relation, follow or unfollow twitterers, as well as query results of interest which can be twitted by the user. This interaction could be handled directly by the machine, but it would conflict with the Automation Rules and Best Practices from Twitter which prevents from automatic (un-)following. In our experiment, we have used a *Companion* account to interact with Twitter according to the recommendation of the machine.

Display of results The results are displayed in the user's browser. We illustrate the behavior of the machine on the query "Hollande", candidate to the French presidential election. On Fig. 2., the first series of tweets, under "Tweets Draft", satisfy a query extracted from Twitter, while the second series, under "Tweets Comp", are tweets produced by twitterers followed by the Companion.

The tweets visualized in Tweets Draft are ordered in decreasing order by twitterers that produced the largest number of tweets satisfying the query since the query was launched. The twitterer together with the number of corresponding tweets can be seen on the display. For the Tweets Comp, the twitterers are displayed together with the proportion of their tweets satisfying the query. Consider for instance the twitterer @LeSecondTour. It is associated with 21 tweets caught by the Tweets Draft, while for the *Companion*, 16 tweets (vs 21) have been caught (we started following this twitterer some time after the query was initially launched), and 87.5% of its tweets satisfy the query.

3 The Twittering Machine plugins

The Twittering Machine is based on plugins injected at run-time that enable the computation of queries. Each plugin is specified with a dedicated descriptive SQL-like specification and the Twittering Machine hosts and schedules the runtime of each plugin. The descriptive language handles streams manipulation. Each stream interaction is triggered with clocks whose parameters are specified within the query. The generic query structure has a form of the following type:

```
Every <X> seconds,
insert into <OutStreamName> values <[Val]>
    Every <Y> seconds,
    compute <[Val] = function()>,
    from <InStreamName>,
    where <Request>.
```

This listing shows the two independent loops of the query. The internal loop queries every <Y> seconds an <InStreamName> stream and maintains an internal array of results [Val]. Then, every <X> seconds, the external loop takes every [Val] and injects them into the <OutStreamName> stream.

From this main structure, we can express a query that produces on the display, every five seconds, an array of the most active twitterers on French election candidate François Hollande.

```
Every 10s,
insert into Console values [twitterers]
  Every 5s,
  [twitterers] = topK (screenname),
  from SearchAPI,
  where tweet like ’%Hollande%’
```

At runtime, when this request is plugged into the Twittering Machine, the scheduling is controlled, such that all streams may not be requested too often. The machine indexes every requested stream and collects timing constraints, if a stream is too much in demand, the twittering machine will lower timing values. Moreover, for performance reasons, if the internal memory for maintaining the array of twitterers is overloaded, the Twittering Machine will reduce its size. Finally, if the Twittering Machine can host the plugin, it compiles the query and generates a equivalent JavaScript code fragment (in the current implementation, the code fragment is generated by hand) that interacts with Twitter. For instance, without considering the topK computation part, the following code is generated for the internal loop.

```
my.run = function (from) {
 $.getJSON(”http://search.twitter.com/search.json?callback=?”,
      {’since_id’:from,’q’:’hollande’,’rpp’:’40’,’lang’:’fr’},
    function(tweets) {
      if (typeof(tweets.results) !== ’undefined’) {
        if(tweets.results.length > 0){
          $.each(tweets.results, function() {
            my.tweetsCpt++;
            DICE.ee.emit(’draft::addTweet’, this);
          });
        }
      }
    }
 setTimeout(function()
        {DICE.draft.run(tweets.max_id_str);},10000);
}
```

The Twittering Machine identifies every input and output as streams. We distinguish between the following streams.

- **Non mutable streams.** Those streams can be both used as input stream such as the main search twitter stream or as output stream as the internal console. These streams are mostly either read-only or write-only from the Twittering Machine perspective.

- **Authenticated streams.** These streams identify user interactions with the external system. For instance the Twittering Machine plugin may express actions such as insert a new follower within a user stream. If the user validates this action the insertion is authenticated.
- **Internal Streams.** These streams mimic an external stream, but for performance, security or privacy reasons they are not available outside the Twittering Machine. For instance, we consider the 'where request' as a flow of query units that can be improved with specific tags. This 'request specification' stream is internally maintained and is not subject to external announcement.

The Twittering Machine manages resources associated to local computations. Each plugin is dynamically deployed within the machine. At deployment time, the framework checks compatibilities with the current installed plugins. As soon as a plugin is validated, the framework monitors its behavior and checks whether it is compliant with Twitter's rules, as well as the amounts of memory and CPU it requires.

The stream SQL like queries describing the plugins are compiled into control behavioral constraints. Each input and output stream pace is dynamically bounded and adapted to the current computer load. Each plugin internal memory is evaluated and adapted when possible to the currently available memory. Our current target implementation uses the web browser javascript interpreter as the Twittering Machine runtime. We designed a plugin based framework that enables dynamic loading and hosting of requests with respect to available resources.

4 Experiments

We have run experiments on tweets related to the French presidential election. As we discussed above, there are many systems giving useful trends on the popularity of candidates, which are now widely used together with traditional opinion polls. They generally extract knowledge from tweets which are essentially seen as raw data, and are often not included in the output. The Twittering Machine on the other hand identifies the relevant tweets and the relevant twitterers, and produces them as output.

We illustrate its behavior on a simple query searching for the keyword "Hollande", the socialist candidate, whose result is shown on Fig. 2. The stream of tweets obtained from Twitter's API is shown as Tweets Draft, while the stream of tweets obtained from followed twitterers is shown as Tweets Comp. The initial query was modified dynamically, with the evolution of the stream of keywords of the query, to which hashtags occurring in the most relevant tweets were added, in the limits permitted by the API.

Interestingly, the results obtained for other candidates overlapped massively with one another, for tweets are often comparative. We did not use any semantical or analytical methods, although they could clearly be combined to our

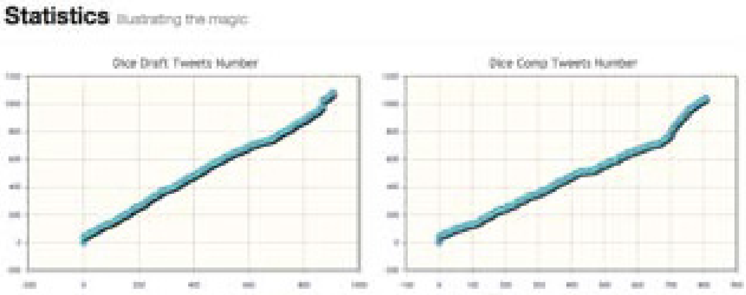


Fig. 3. Statistical information displayed

approach, but it is out of the scope of this paper. The Twittering Machine produces real time statistics on the tweets satisfying the query, which are shown on Fig. 3. The curve on the left depends upon Twitter's search API, while the one on the right depends upon the twitterers followed. Interestingly, the two curves become very similar with time, showing the relevance of the choices of the machine.

Among the interesting results that this query allowed to grasp, was the hashtag #radioLondres, which was used by twitterers to publish results on the first round of the election before it was legally allowed in France.

5 Related Work

There has been an exponential increase of academic work on techniques to extract knowledge from the data contained in tweets, while online systems to analyze them have flourished. The simplest systems offer graphical interfaces to watch the tweets themselves, their numbers, or relative numbers, and their dynamics. Semiocast[5] for instance produces a daily or monthly ranking of, for instance, major French politicians on Twitter with associated mood. It harvests the tweets continuously using queries on a set of keywords that can be updated, depending upon current affairs or evolving nicknames. It disambiguates them, and analyses their mood, and produces charts that are easily readable.

Twitter offers also various access to trends, which unlike Google trends, belong to the top trends only. They have been graphically organized on maps for instance with tools such as Trendsmap[6] which displays them on the world map. Apart from France, search for Sarkozy gave (on April 30) results in Jakarta, Kuala Lumpur and Brisbane, showing the sometimes awkward results of these tools. An equivalent of Flu Trend [AGL+11] has been developed based on similar principle as the one developed by Google. This illustrates the potential of Twitter whose tweets can be easily accessed, while the search queries on Google are out of reach of users and used exclusively by Google for commercial purposes.

[5] http://semiocast.com/barometre_politique

[6] http://trendsmap.com/

Many systems rely on complex analysis of tweet data, including statistics, semantics, etc. This is the main focus of most of the papers of the workshop devoted to Making Sense of Microposts [RSD12]. Thompson [Tho12] defines a stochastic model to detect anomalies in streaming communication data that is applicable to information retrieval in data streams. Some systems also measure the influence, e.g. Klout[7], or the value in financial terms of a twitterer, e.g. Tweetvalue[8].

Our aim is to offer the capacity to find relevant tweets on any topic, satisfying any type of queries. Two issues are thus at stake, the query language and the ranking. Stream queries have attracted a lot of attention in the last ten years, before their use for social data became important, with SQL like languages [MWA+03], formalized in [GLdB07]. Markus et al. developed a streaming SQL-like interface, TweeQL, to the Twitter API [MBB+11a, MBB+11b]. The language in the Twittering Machine, differs essentially from TweeQL, for its richer interactions with Twitter. We also assume that the continuous queries need to evolve over time. This topic has been considered recently in [ESFT11].

Adaptive indexing mechanisms for tweets have been proposed distinguishing between most frequent query terms [CLOW11]. Ranking is as for other web data an important issue, which combines content information with graph connections, and requires a recursive computation. Mechanisms to rank tweets based on pagerank like techniques have been proposed[9]. Topic-sensitive ranking algorithm which take users interest into account [Hav03], have been considered for tweets as well [KF11]. Variants of ranking can be considered in the Twittering Machine, whose first version implements a simple algorithm computed at query time. Various notions of relevance have been introduced. Tao et al. [TAHH12] propose to combine topic independent aspects, such as hashtags, URLs, and authorities of the twitterers.

6 Conclusion

Extracting knowledge from tweets has attracted considerable interest thanks to the generous access Twitter gives through its API. Nevertheless, most tweets are related to notification substance, not always of immediate interest. Identifying relevant tweets and twitterers on a given topic, and not only for top trends is of great importance. In this paper we have presented the Twittering Machine which allows to get tweets according to a ranking, which is computed in a continuous manner by a query engine which runs a stream SQL like query language.

The Twittering Machine, coded in javascript, runs directly in the Browser. It requires essentially no server, runs on the client and relies directly on the API of Twitter. We believe that this server-less approach is extremely promising. We plan to extend the present machine to allow collaboration between users interested in similar queries. The objective is to maintain the computation of

[7] http://klout.com/home

[8] http://tweetvalue.com/

[9] http://thenoisychannel.com/2009/01/13/a-twitter-analog-to-pagerank/

queries, by dynamic groups of users, that are organized in a peer to peer fashion, and contribute to the update of the output stream of the query when they consult it, much like a torrent. This *query torrent*, will rely on the cooperation of peers for the computation of queries, with therefore more computation involved than file torrents. This approach raises security issues related in particular to communication between browsers that we are investigating.

Acknowledgements. The authors would like to thank Fabien Culpo, who participated in the implementation of the Twittering Machine.

References

[AGL$^+$11] Achrekar, H., Gandhe, A., Lazarus, R., Yu, S.-H., Liu, B.: Predicting flu trends using twitter data. In: IEEE Conference on Computer Communications Workshops, INFOCOM (2011)

[BGL$^+$12] Busch, M., Gade, K., Larson, B., Lok, P., Luckenbill, S., Lin, J.: Earlybird: Real-time search at twitter. In: IEEE International Conference on Data Engineering, ICDE (2012)

[CLOW11] Chen, C., Li, F., Ooi, B.C., Wu, S.: Ti: an efficient indexing mechanism for real-time search on tweets. In: ACM SIGMOD International Conference on Management of Data, Athens (2011)

[ESFT11] Esmaili, K.S., Sanamrad, T., Fischer, P.M., Tatbul, N.: Changing flights in mid-air: a model for safely modifying continuous queries. In: ACM SIGMOD International Conference on Management of Data, Athens (2011)

[GLdB07] Gurevich, Y., Leinders, D., Van den Bussche, J.: A theory of stream queries. In: 11th International Symposium on Database Programming Languages, DBPL, Vienna (2007)

[GMP$^+$09] Ginsberg, J., Mohebbi, M., Patel, R., Brammer, L., Smolinski, M., Brilliant, L.: Detecting influenza epidemics using search engine query data. Nature 457, 1012–1014 (2009)

[Hav03] Haveliwala, T.H.: Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. IEEE Trans. Knowl. Data Eng. 15(4), 784–796 (2003)

[KF11] Kong, S., Feng, L.: A Tweet-Centric Approach for Topic-Specific Author Ranking in Micro-Blog. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS, vol. 7120, pp. 138–151. Springer, Heidelberg (2011)

[MBB$^+$11a] Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Processing and visualizing the data in tweets. SIGMOD Record 40(4), 21–27 (2011)

[MBB$^+$11b] Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Tweets as data: demonstration of tweeql and twitinfo. In: ACM SIGMOD International Conference on Management of Data (2011)

[MWA$^+$03] Motwani, R., Widom, J., Arasu, A., Babcock, B., Babu, S., Datar, M., Manku, G.S., Olston, C., Rosenstein, J., Varma, R.: Query processing, approximation, and resource management in a data stream management system. In: CIDR (2003)

[RSD12] Rowe, M., Stankovic, M., Dadzie, A.-S. (eds.): Proceedings, 2nd Workshop on Making Sense of Microposts (#MSM 2012): Big things come in small packages, Lyon, France, April 16 (2012)

[TAHH12] Tao, K., Abel, F., Hauff, C., Houben, G.-J.: What makes a tweet relevant for a topic? In Rowe et al. [RSD 12], pp. 49–56

[Tho12] Thompson, B.: The early bird gets the buzz: detecting anomalies and emerging trends in information networks. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM 2012 (2012)

Contextual Recommendations for Groups

Kostas Stefanidis*, Nafiseh Shabib, Kjetil Nørvåg, and John Krogstie

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Trondheim, Norway
{kstef,shabib,kjetil.norvag,krogstie}@idi.ntnu.no

Abstract. Recommendation systems have received significant attention, with most of the proposed methods focusing on recommendations for single users. Recently, there are also approaches aiming at either group or context-aware recommendations. In this paper, we address the problem of *contextual recommendations for groups*. We exploit a hierarchical context model to extend a typical recommendation model to a general context-aware one that tackles the information needs of a group. We base the computation of contextual group recommendations on a subset of preferences of the users that present the most similar behavior to the group, that is, the users with the most similar preferences to the preferences of the group members, for a specific context. This subset of preferences includes the ones with context equal to or more general than the given context.

1 Introduction

Recommendation systems provide users with suggestions about products, movies, videos, pictures and many other items. Many systems, such as Amazon, NetFlix and MovieLens, have become very popular. Typically, recommendation approaches are distinguished between: *content-based*, that recommend items similar to those the user previously preferred (e.g., [20,15]), *collaborative filtering*, that recommend items that users with similar preferences liked (e.g., [13,8]) and *hybrid*, that combine content-based and collaborative ones (e.g., [3,5]).

The two types of entities that are dealt with in recommendation systems, i.e., users and items, are represented as sets of ratings, preferences or features. Assume, for example, a restaurant recommendation application (e.g., ZAGAT.com). Users initially rate a subset of restaurants that they have already visited. Ratings are expressed in the form of preference scores. Then, a recommendation engine estimates preference scores for the items, e.g., restaurants, that are not rated by a user and offers appropriate recommendations. Once the unknown scores are computed, the k items with the highest scores are recommended to the user.

* This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 89–97, 2012.

Since recommendations are typically personalized, different users are presented with different suggestions. However, there are cases in which a group of people participates in a single activity. For instance, visiting a restaurant or a tourist attraction, watching a movie or a TV program and selecting a holiday destination are examples of recommendations well suited for groups of people. For this reason, recently, there are approaches addressing the problem of identifying recommendations for groups, trying to satisfy the preferences of all the group members (e.g., [18,2,4,16]).

Moreover, often users have different preferences under different circumstances. For instance, the current weather conditions may influence the place one wants to visit. For example, when it rains, a museum may be preferred over an open-air archaeological site. *Context* is a general term used in several domains, such as in machine learning and knowledge acquisition [9,6]. Our focus here is on how context can be used in conjunction with recommendation systems. In this respect, we consider as context any information that can be used to characterize the situations of an entity, where an entity is a person, place, or object that is relevant to the interaction between a user and an application [11]. Common types of context include the computing context (e.g., network connectivity, nearby resources), the user context (e.g., profile, location), the physical context (e.g., noise levels, temperature) and time [10,7]. Several approaches, such as [1] and [19], extend the typical recommendation systems beyond the two dimensions of users and items to include further contextual information.

In this paper, we address the problem of *contextual recommendations for groups*. In general, different approaches have been proposed in the research literature focusing on either group or context-aware recommendations. However, as far as we know, this is the first work presenting a complete model for contextual recommendations for groups. The context model of our previous work [23] serves as a building brick for extending a typical recommendation model to a general context-aware one that tackles the information needs of a group.

The computation of contextual group recommendations proceeds in the following main phases. First, given a group of users along with a context state, or situation, we locate the users that exhibit the most similar behavior or, in other words, have expressed the most similar preferences, to the group for the given context. We call such users *peers* of the group. Next, we employ the peers preferences defined for the given context to identify the items to be suggested to the group. Since many times there are no or not enough preferences for a specific context, we consider also issues underlying context relaxation and so, employ preferences with context more general than the given one.

The rest of the paper is organized as follows. Sect. 2 presents our context model, as well as our model for contextual single user and group recommendations. Sect. 3 focuses on the three main phases for computing contextual group recommendations, namely, (i) peers selection, (ii) preferences selection and (iii) recommendations computation. Finally, Sect. 4 draws conclusions and future work.

2 A Contextual Group Recommendation Model

Assume a set of items $\mathcal{I}$ and a set of users $\mathcal{U}$ interacting with a recommendation application. Each user $u \in \mathcal{U}$ may express, for a context state cs, a contextual preference for an item $i \in \mathcal{I}$, which is denoted by $cpref(u, i, cs)$ and lies in the range $[0.0, 1.0]$. As a running example, we shall use a movie recommendation application.

In the following, we first present our context model and then focus on the specification of a contextual recommendation model for single users and groups.

2.1 Context Model

A variety of models for context have been proposed (see, for example, [21] for a survey). We follow the data-centric approach of [23]. Context is modeled as a set of n context parameters $C_1, \ldots, C_n$, where each C_i, $1 \leq i \leq n$, captures information that is not part of the database, such as the user location, the current weather or time. For our movie example, let us assume that context consists of *Weather* and *Time period*. Each context parameter takes values from a hierarchical domain, so that different levels of abstraction for the captured context data are introduced.

In particular, each context parameter has multiple levels organized in a hierarchy schema. Let C be a context parameter with $m > 1$ levels, L_i, $1 \leq i \leq m$. We denote its hierarchy schema as $L = (L_1, ..., L_m)$. L_1 is called the lowest or most detailed level of the hierarchy schema and L_m the top or most general one. We define a total order among the levels of L such that $L_1 \prec \ldots \prec L_m$ and use the notation $L_i \preceq L_j$ between two levels to mean $L_i \prec L_j$ or $L_i = L_j$. Fig. 1 depicts the hierarchy schemas of the context parameters of our running example. For instance, the hierarchy schema of context parameter *Time period* has three levels: *occasion* (L_1), *interval* (L_2) and the top level *ALL* (L_3). Each level L_j, $1 \leq j \leq m$, is associated with a domain of values, denoted $dom_{L_j}(C)$. For all parameters, their top level has a single value *All*, i.e., $dom_{L_m}(C) = \{All\}$. A concept hierarchy is an instance of a hierarchy schema, where the concept hierarchy of a context parameter C with m levels is represented by a tree with m levels with nodes at each level j, $1 \leq j \leq m$, representing values in $dom_{L_j}(C)$. The root node (i.e., level m) represents the value *All*. Fig. 1 depicts the concept hierarchies of the context parameters of our running example. For instance, for the context parameter *Time period*, *holidays* is a value of level *interval*. The relationship between the values at the different levels of a concept hierarchy is achieved through the use of a family of ancestor and descendant functions [24]. Finally, we define the domain, $dom(C)$, of C as: $dom(C) = \cup_{j=1}^{m} dom_{L_j}(C)$.

A context state cs is defined as an n-tuple $(c_1, \ldots, c_n)$, where $c_i \in dom(C_i)$, $1 \leq i \leq n$. For instance, (*warm, holidays*) and (*cold, weekend*) are context states for our movie example. The set of all possible context states, called world CW, is the Cartesian product of the domains of the context parameters: $CW = dom(C_1) \times \ldots \times dom(C_n)$.

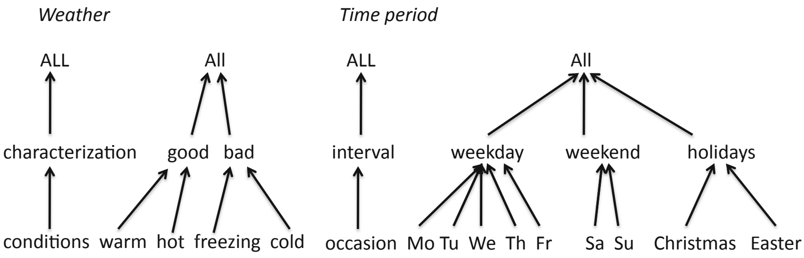


Fig. 1. Hierarchy schema and concept hierarchy of *Weather* and *Time period*.

2.2 Contextual Recommendations for Single Users

In general, there are different ways to estimate the relevance of an item for a user under a specific context state by employing a set of available user contextual preferences of the form $cpref(u, i, cs)$. The meaning of such a preference is that in the context state specified by cs, the movie, in our case, i was rated by user u with a score. For example, according to $cpef$(*Tim, The Hangover,* (*warm, weekend*)) = 0.8, *Tim* gave a high rate to the comedy movie *The Hangover* at a *warm weekend*, while $cpef$(*Alice, Toy Story,* (*cold, Christmas*)) = 0.9 defines that *Alice* likes the animation movie *Toy Story* during *cold Christmas.*

Our work falls into the *collaborative filtering* category. The key concept of collaborative filtering is to use preferences of other users that exhibit the most similar behavior to a given user, for a specific context, in order to predict relevance scores for unrated items. Similar users are located via a *similarity function* $simU_{cs}(u, u')$, that evaluates the proximity between u and u' for cs.

We use $\mathcal{P}_{u,cs}$ to denote the set of the most similar users to u for a context state cs. Or, in other words, the users with preferences similar to the preferences of u for cs. We refer to such users as the *peers* of u for cs. Several methods can be employed for selecting $\mathcal{P}_{u,cs}$. A direct method is to locate those users u' with similarity $simU_{cs}(u, u')$ greater than a threshold value. This is the method used in this work. Formally, peers are defined as follows:

Definition 1 (Peers of a Single User). *Let $\mathcal{U}$ be a set of users. The peers $\mathcal{P}_{u,cs}$, $\mathcal{P}_{u,cs} \subseteq \mathcal{U}$, of a user $u \in \mathcal{U}$, for a context state cs, is a set of users, such that, $\forall u' \in \mathcal{P}_{u,cs}$, $simU_{cs}(u, u') \geq \delta$ and $\forall u'' \in \mathcal{U} \backslash \mathcal{P}_{u,cs}$, $simU_{cs}(u, u'') < \delta$, where δ is a threshold similarity value.*

Clearly, one could argue for other ways of selecting $\mathcal{P}_{u,cs}$, for instance, by taking the k most similar users to u. Our main motivation is that we opt for selecting only highly connected users even if the resulting set of users $\mathcal{P}_{u,cs}$ is small.

Next, we define the contextual relevance of an item recommendation for a user.

Definition 2 (Single User Contextual Relevance). *Given a user $u \in \mathcal{U}$ and his peers $\mathcal{P}_{u,cs}$ for a context state cs, the single user contextual relevance of* *an item $i \in \mathcal{I}$ for u under cs, such that, $\nexists cpref(u, i, cs)$, is:*

$$crel(u, i, cs) = \frac{\sum_{u' \in (\mathcal{P}_{u,cs} \cap \mathcal{X}_{i,cs})} simU_{cs}(u, u')cpref(u', i, cs)}{\sum_{u' \in (\mathcal{P}_{u,cs} \cap \mathcal{X}_{i,cs})} simU_{cs}(u, u')}$$

where $\mathcal{X}_{i,cs}$ is the set of users in $\mathcal{U}$ that have expressed a preference for item i for context state cs.

2.3 Contextual Recommendations for Groups

The large majority of recommendation systems are designed to make personal recommendations, i.e., recommendations for single users. However, there are cases in which the items to be selected are not intended for personal usage but for a group of users. For example, assume a group of friends or a family that is planning to watch a movie together. Existing methods to construct a ranked list of recommendations for a group of users can be classified into two approaches [12]. The first approach aggregates the recommendations of each user into a single recommendation list (e.g., [2,4,17]), while the second one creates a joint profile for all users in the group and provides the group with recommendations computed with respect to this joint profile (e.g., [14,25]).

In our work, we adopt the second approach to offer context-aware recommendations to groups. The first step towards this direction is to locate the similar users to the group, whose preferences will be used for making suggestions.

For a context state *cs*, we define the similarity between a user u and a group of users $\mathcal{G}$ as follows:

$$simG_{cs}(u, \mathcal{G}) = Aggr_{u' \in \mathcal{G}}(simU_{cs}(u, u'))$$

We employ two different designs regarding the aggregation method *Aggr*, each one carrying different semantics: (i) the *least misery design*, where the similarity between the user u and the group $\mathcal{G}$ is equal to the minimum similarity between u and any other user in $\mathcal{G}$, and (ii) the *fair design*, where the similarity between u and $\mathcal{G}$ is equal to the average similarity between u and all users in $\mathcal{G}$. The least misery design captures cases where strong user preferences act as a veto, e.g., do not recommend thriller movies to a group when a group member extremely dislike them, while the fair design captures more democratic cases where the majority of the group is satisfied.

Then, the peers of a group for a context state *cs* are defined as:

Definition 3 (Peers of a Group). *Let $\mathcal{U}$ be a set of users. The peers $\mathcal{P}_{G,cs}$, $\mathcal{P}_{G,cs} \subseteq \mathcal{U}$, of a group $\mathcal{G}$, for a context state cs, is a set of users, such that, $\forall u \in \mathcal{P}_{G,cs}$, $simG_{cs}(u, \mathcal{G}) \geq \delta'$ and $\forall u' \in \mathcal{U} \backslash \mathcal{P}_{G,cs}$, $simG_{cs}(u', \mathcal{G}) < \delta'$, where δ' is a threshold similarity value.*

Based on the notion of peers for a group, we define next the contextual relevance of an item for a group for a specific context state.

Definition 4 (Group Contextual Relevance). *Given a group $\mathcal{G}$ and its peers $\mathcal{P}_{\mathcal{G},cs}$ for a context state cs, the group contextual relevance of an item $i \in \mathcal{I}$ for $\mathcal{G}$ under cs, such that, $\forall u \in \mathcal{G}$, $\nexists cpref(u, i, cs)$, is:*

$$crel(\mathcal{G}, i, cs) = \frac{\sum_{u \in (\mathcal{P}_{\mathcal{G},cs} \cap \mathcal{X}_{i,cs})} simG_{cs}(u, \mathcal{G}) cpref(u, i, cs)}{\sum_{u \in (\mathcal{P}_{\mathcal{G},cs} \cap \mathcal{X}_{i,cs})} simG_{cs}(u, \mathcal{G})}$$

where $\mathcal{X}_{i,cs}$ is the set of users in $\mathcal{U}$ that have expressed a preference for item i for context state cs.

3 Computing Contextual Group Recommendations

A high level representation of the main components of the architecture of our system is depicted in Fig. 2. First, a group poses a query presenting its information needs. Each query is enhanced with a contextual specification, that is, a context state denoted as cs^Q. The context of the query may be postulated by the application or be explicitly provided by the group as part of the query. Typically, in the first case, the context associated with a query corresponds to the current context, that is, the context surrounding the group at the time of the submission of the query. Such information may be captured using appropriate devices and mechanisms, such as temperature sensors or GPS-enabled devices for location. Besides this implicit context, a group may explicitly specify a context state. For example, assume a group that expresses an exploratory query asking for interesting movies to watch over the coming *cold weekend*.

Given a specific query, computing contextual group recommendations involves three phases: peers selection, preferences selection and recommendations computation. In following, we describe each of these phases in detail.

Peers Selection. For locating the peers of a group $\mathcal{G}$ for a context state cs, we need to calculate the similarity measures $simG_{cs}(u, \mathcal{G})$, $\forall u \in \mathcal{U} \backslash \mathcal{G}$. Those users u with similarity $simG_{cs}(u, \mathcal{G})$ greater than δ' represent the peers of $\mathcal{G}$ for cs, $\mathcal{P}_{G,cs}$.

The notion of user similarity is important, since it determines the produced peers set. We use here a simple variation; that is, we use distance instead of similarity. More specifically, we define the distance between two users as the Euclidean distance over the items rated by both under the same context state. Let $u, u' \in \mathcal{U}$ be two users, $\mathcal{I}_u$ be the set of items for which $\exists cpref(u, i, cs)$, $\forall i \in \mathcal{I}_u$, and $\mathcal{I}_{u'}$ be the set of items for which $\exists cpref(u', i, cs)$, $\forall i \in \mathcal{I}_{u'}$. We denote by $\mathcal{I}_u \cap \mathcal{I}_{u'}$ the set of items for which both users have expressed preferences for cs. Then, the distance between u, u' is defined as:

$$distU_{cs}(u, u') = \frac{\sqrt{\sum_{i \in \mathcal{I}_u \cap \mathcal{I}_{u'}} (cpref(u, i, cs) - cpref(u', i, cs))^2}}{|\mathcal{I}_u \cap \mathcal{I}_{u'}|}$$

The similarity between two users, $simU_{cs}(u, u')$, is equal to $1 - distU_{cs}(u, u')$, based on which the similarity between a user and a group is computed.

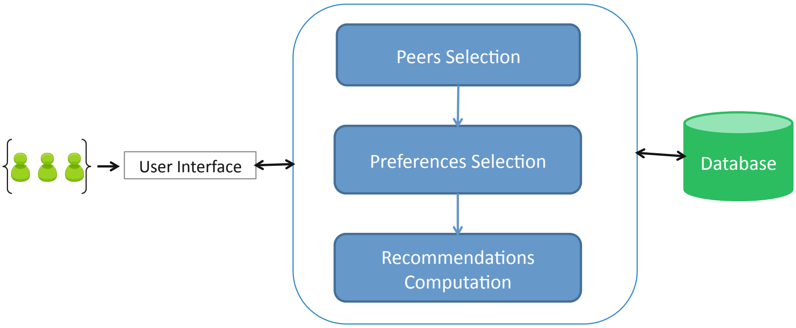


Fig. 2. System architecture

Preferences Selection. Given a group $\mathcal{G}$, preferences selection determines which preferences from the peers of $\mathcal{G}$ will be employed for making recommendations. For example, assume that a group wants to find a movie to watch on a *Sunday*. Then, the peers preferences for *weekdays* are outside the query context.

Clearly, in terms of context, a preference $cpref(u, i, cs)$ can be used if cs is equal to the query context cs^Q. However, when there are no such peers preferences, or when their number is small, we may need to select, in addition, preferences whose context state is not necessarily the same with cs^Q, but close enough to it. To determine how close the preference and query contexts are, we rely on an appropriate distance measure. Since our context parameters take values from hierarchical domains, we exploit this fact and relate contexts expressed at different levels of detail. For instance, we can relate a context in which the parameter *Time period* is instantiated to a specific occasion (e.g., *Christmas*) with a context in which the same parameter describes a more general period (e.g., *holidays*). Intuitively, a preference defined for a more general value, e.g., *holidays*, may be considered applicable to a query about a more specific one, e.g., *Christmas*. In general, we relate the context of a preference to the context of a query, if the first one is more general than the second, that is, if the context values specified in cs are equal to or more general than the ones specified in cs^Q. In this case, we say that the preference context *covers* the query context [23].

Given a preference state $cs = (c_1, \ldots, c_n)$ and a query state $cs^Q = (c_1^Q, \ldots, c_n^Q)$, where cs covers cs^Q, we quantify their relevance based on how far away are their values in the corresponding hierarchies.

$$dist_H(cs, cs^Q) = \sum_{i=1}^{n} d_H(level(c_i),\ level(c_i^Q)),$$

where $level(c_i)$ (resp. $level(c_i^Q)$) is the hierarchy level of value c_i (resp. c_i^Q) of parameter C_i and d_H is equal to the number of edges that connect $level(c_i)$ and $level(c_i^Q)$ in the hierarchy of C_i, $1 \leq i \leq n$.

[22] studies different ways of relaxing context. In particular, a context parameter can be relaxed *upwards* by replacing its value by a more general one,

downwards by replacing its value by a set of more specific ones or *sideways* by replacing its value by sibling values in the hierarchy.

The output of this phase is a set of m preferences with contexts closest to the query context, sorted on the basis of their distance to cs^Q, from the set of preferences of the users in $\mathcal{P}_{G,cs}$.

Recommandations Computation. For estimating the value of an item i for a group $\mathcal{G}$ under a context state cs, we compute its group contextual relevance $crel(\mathcal{G}, i, cs)$ (Def. 4), taking into account the output of the previous phase. We do not compute scores for all items in $\mathcal{I}$, but only for the items $\mathcal{I}'$, $\mathcal{I}' \subseteq \mathcal{I}$, that satisfy the selection conditions of the posed query. As a post-processing step, we rank the items in $\mathcal{I}'$ on the basis of their score and report the k items with the highest scores.

4 Conclusions

The focus of this paper is on contextual recommendations for groups. Context is modeled using a set of context parameters that take values from hierarchical domains. A context state corresponds to an assignment of values to each of the context parameters from its corresponding domain. User preferences are augmented with context states that specify the situations under which preferences hold. Given a group of users associated with a context state, we consider the problem of providing the group with context-aware recommendations. To do this, we follow a collaborative filtering approach that uses the preferences of the similar users to the group members defined for the context surrounding the group at the time of recommendations computation or any other explicitly defined context.

In our current work, we are developing a Java prototype to add a capability for producing contextual group recommendations to our group recommendations system [17]. There are many directions for future work. One is to extend our model so as to support additional ways for locating the peers of a group of users. Another direction for future work is to consider recency issues when computing contextual recommendations. For example, since usually the most recent user preferences reflect better the current trends, it is promising to examine if they should contribute more in the computation of the contextual group recommendations.

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. 23(1), 103–145 (2005)
2. Amer-Yahia, S., Roy, S.B., Chawla, A., Das, G., Yu, C.: Group recommendation: Semantics and efficiency. PVLDB 2(1), 754–765 (2009)
3. Balabanovic, M., Shoham, Y.: Content-based, collaborative recommendation. Commun. ACM 40(3), 66–72 (1997)
4. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: RecSys, pp. 119–126 (2010)

5. Basu, C., Hirsh, H., Cohen, W.W.: Recommendation as classification: Using social and content-based information in recommendation. In: AAAI/IAAI, pp. 714–720 (1998)
6. Bolchini, C., Curino, C., Orsi, G., Quintarelli, E., Rossato, R., Schreiber, F.A., Tanca, L.: And what can context do for data? Commun. ACM 52(11), 136–140 (2009)
7. Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F.A., Tanca, L.: A data-oriented survey of context models. SIGMOD Rec. 36(4), 19–26 (2007)
8. Breese, J.S., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI, pp. 43–52 (1998)
9. Brézillon, P.: Context in artificial intelligence: I. a survey of the literature. Computers and Artificial Intelligence 18(4) (1999)
10. Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Dartmouth College, Computer Science (November 2000)
11. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput 5(1), 4–7 (2001)
12. Jameson, A., Smyth, B.: Recommendation to Groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)
13. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: Grouplens: Applying collaborative filtering to usenet news. Commun. ACM 40(3), 77–87 (1997)
14. McCarthy, J.F., Anagnost, T.D.: Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In: CSCW, p. 348 (2000)
15. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: ACM DL, pp. 195–204 (2000)
16. Ntoutsi, E., Stefanidis, K., Nørvåg, K., Kriegel, H.-P.: Fast Group Recommendations by Applying User Clustering. In: Atzeni, P., Cheung, D., Sudha, R. (eds.) ER 2012. LNCS, vol. 7532, pp. 126–140. Springer, Heidelberg (2012)
17. Ntoutsi, I., Stefanidis, K., Norvag, K., Kriegel, H.-P.: gRecs: A Group Recommendation System Based on User Clustering. In: Lee, S.-g., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part II. LNCS, vol. 7239, pp. 299–303. Springer, Heidelberg (2012)
18. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: A recommender system for groups of user. In: ECSCW, pp. 199–218 (2001)
19. Palmisano, C., Tuzhilin, A., Gorgoglione, M.: Using context to improve predictive modeling of customers in personalization applications. IEEE Trans. Knowl. Data Eng. 20(11), 1535–1549 (2008)
20. Pazzani, M.J., Billsus, D.: Learning and revising user profiles: The identification of interesting web sites. Machine Learning 27(3), 313–331 (1997)
21. Stefanidis, K., Koutrika, G., Pitoura, E.: A survey on representation, composition and application of preferences in database systems. ACM Trans. Database Syst. 36(3), 19 (2011)
22. Stefanidis, K., Pitoura, E., Vassiliadis, P.: On relaxing contextual preference queries. In: MDM, pp. 289–293 (2007)
23. Stefanidis, K., Pitoura, E., Vassiliadis, P.: Managing contextual preferences. Inf. Syst. 36(8), 1158–1180 (2011)
24. Vassiliadis, P., Skiadopoulos, S.: Modelling and Optimisation Issues for Multidimensional Databases. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 482–497. Springer, Heidelberg (2000)
25. Yu, Z., Zhou, X., Hao, Y., Gu, J.: Tv program recommendation for multiple viewers based on user profile merging. User Model. User-Adapt. Interact. 16(1), 63–82 (2006)

First International Workshop on Modeling for Data-Intensive Computing

Preface

Due to the enormous amount of data present and growing in the Web, there has been an increasing interest in incorporating the huge amount of external and unstructured data, normally referred as "Big Data", into traditional applications. This necessity has made that traditional database systems and processing need to evolve and accommodate them to this new situation. Two main ideas underneath this evolution are that this new external and internal data (i) need to be stored in the cloud and (ii) offer a set of services to allow us to access, abstract, analyze, and visualize the data.

Therefore, this new conceptualization of cloud applications incorporating both internal and external Big Data requires new models and methods to accomplish their conceptual modeling phase. Thus, the objective of the First International Workshop on Modeling for Data-Intensive Computing (MoDIC'12) is to be an international forum for exchanging ideas on the latest and best proposals for the conceptual modeling issues surrounding this new data-drive paradigm with Big Data. Papers focusing on the application and the use of conceptual modeling approaches for Big Data, MapReduce, Hadoop and Hive, Big Data Analytics, social networking, security and privacy data science, etc. will be highly encouraged. The workshop will be a forum for researchers and practitioners who are interested in the different facets related to the use of the conceptual modeling approaches for the development of this next generation of applications based on these Big Data.

The workshop attracted papers from 9 different countries distributed all over the world: Brazil, China, Cuba, France, Poland, Spain, Sweden, Switzerland and USA. We received 12 papers and the Program Committee selected only four papers, making an acceptance rate of 33.3%. We also have an invited paper entitled A Comprehensive Model for Provenance by Sultana, S. and Bertino, E.; and an invited keynote on the quality of big and scientific data given by Carlo Batini.

The different talks are organized in two sessions. The first one will be focused on Big Data: general issues and modeling approaches, including the keynote speech and two accepted papers. The second session includes the invited paper and two accepted papers.

We would like to express our gratitude to the Program Committee members for their hard work in reviewing papers, the authors for submitting their papers, and the ER 2012 organizing committee for all their support.

June 2012

David Gil
Juan Trujillo
Il-Yeol Song
Program Co-Chairs
MoDIC'12

A Scientific Hypothesis Conceptual Model

Fabio Porto[1], Ana Maria de C. Moura[1], Bernardo Gonçalves[1],
Ramon Costa[1], and Stefano Spaccapietra[2]

[1] LNCC – National Laboratory of Scientific Computing, DEXL – Extreme Data Lab.,
Petropolis – Brazil
{fporto,anamoura,bgonc,ramongc}@lncc.br
[2] EPFL – IC – LBD, Lausanne, Switzerland
stefano.spaccapietra@epfl.ch

Abstract. *In-silico* scientific research is a complex task that involves the management of huge volumes of data and metadata produced during the scientific exploration life cycle, from hypothesis formulation up to its final validation. This wealth of data needs to be structured and managed in a way that readily makes sense to scientists, so that relevant knowledge may be extracted to contribute to the scientific investigation process. This paper proposes a scientific hypothesis conceptual model that allows scientists to represent the phenomenon been investigated, the hypotheses formulated in the attempt to explain it, and provides the ability to store results of experiment simulations with their corresponding provenance metadata. The proposed model supports scientific lifecycle: provenance, scientists exchange of information, experiment reproducibility, model steering and results analyses. A cardiovascular numerical simulation illustrates the applicability of the model and an initial implementation using SciDB is discussed.

Keywords: Conceptual Model, eScience, Scientific Hypothesis.

1 Introduction

The availability of important experimental and computational facilities nowadays induces large-scale scientific projects to produce a never before observed amount of experimental and simulation data. This wealth of data needs to be structured and managed in a way that readily makes sense to scientists, so that relevant knowledge may be extracted to contribute to the scientific investigation process. Current data management technologies are clearly unable to cope with scientists' requirements [19], despite the efforts the community has dedicated to the area. Such efforts can be measured by the community support to an international conference (SSDBM), running for almost 20 years on scientific and statistical database management, various workshops on associated themes, and important projects such as POSTGRES at Berkeley [18]. All these initiatives have considerably contributed to extend database technology towards the support to scientific data management.

Giving such a panorama, one may ask what could be missing on the support to scientific applications from a database viewpoint. In this paper, we investigate this

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 101–110, 2012.

question from the perspective of data management support for the complete scientific life-cycle, from hypotheses formulation to experiment validation. As it turns out, efforts in this area have been steered towards supporting the *in-silico* experimental phase of the scientific life-cycle [8], involving the execution of scientific workflows and the management of the associated data and metadata. The complete scientific life-cycle extends beyond that, and includes the studied phenomenon, formulated hypotheses and mathematical models. The lack of support to these elements in current *in-silico* approaches leaves extremely important information out-of-reach of the scientific community.

This paper contributes to fill this gap, by introducing a scientific hypothesis conceptual model. In this model, the starting point of a scientific investigation is the natural phenomenon description. The studied phenomenon occurs in nature in some space-time frame, in which selected physical quantities are observed. Scientific hypotheses conceptually represent the mathematical models a scientist conceives to explain the observed phenomenon. Testing hypotheses *in-silico* involves running experiments, representing the mathematical models, and confronting simulated data with collected observations.

The proposed conceptual model is the basis for registering the complete scientific exploration life-cycle. The following benefits are brought by this approach:

- Extends the *in-silico* support beyond the experimental phase and towards the complete scientific life-cycle;
- Supports provenance information regarding scientific hypotheses evolution;
- Facilitates the communication among scientists in a research groups (by exposing their mental models);
- Supports the reproducibility of experiments (by enhancing the experiment metadata with hypotheses and models);
- Supports model steering (by investigating models evolution);
- Supports experiment result analyses (by relating models, models parameters and simulated results);

In order to illustrate the use of the proposed conceptual model, a case study is discussed, based on models of the human cardio-vascular system. The phenomenon is simulated by a complex and data intensive numerical simulation that runs for days to compute a single blood cycle on a cluster with 1200 nodes. The analyses of simulated results are supported by the SciDB [5].

The remainder of this paper is structured as follows. Section 2 discusses related work. Section 3 describes a use case concerning the simulation of the human cardiovascular system. Section 4 presents the Hypothesis Conceptual Model that integrates scientific hypotheses to the in-silico experiment entities. Section 5 describes a database prototype developed using SciDB in support of the cardio vascular scientific hypothesis. Finally, section 6 concludes the paper with suggestions for future work.

2 Related Work

Data and knowledge management supporting *in-silico* scientific research is a comprehensive topic. It encompasses the semantic description of the scientific domain, experiment evaluation using scientific workflow systems and result management and analysis by means of a myriad of different techniques, in addition to other *in-silico* based research activities. The combination of hardware, software and data put together in support to scientific research has been labelled eScience.

A few areas of research associated to data and knowledge management have, nevertheless, found in eScience a fertile soil for development. The semantic description of scientific domains through ontologies [7] is one such area as a means to support scientific collaboration through common conceptual agreement. In this line, GeneOntology[1] is probably the most well known and successful example of practical adoption of ontologies by the scientific domain. Similarly, scientific workflows have become the de facto standard for expressing and running in-silico experiments, using execution environments, such as Taverna [9], Kepler [1] and QEF [14].

The problem of systematically capturing and managing provenance information for *in-silico* experiments has also attracted significant attention. According to Freire et al.[6], provenance information helps interpreting and understanding results by examining the sequence of steps that led to them. The analysis of provenance data helps verifying that experiments were conducted according to acceptable procedures and eventually supports reproducibility of the experiment. Provenance information has been classified into two categories: prospective and retrospective [4]. The former refers to the description of the experiment design, whereas the latter refers to information used and produced during the *in-silico* experiment evaluation.

Conceptually, this work extends experiment prospective provenance with scientific hypotheses and models. By considering information about the phenomenon, the scientific hypotheses, and models, a more complete description of the problem-hypotheses the experiment tries to evaluate is provided.

The introduction of spatio-temporal and multi-representation in conceptual modelling was the focus of the MADS approach [10]. In MADS, space, time and space-time are organized into class hierarchies to be extended by spatial objects. In computational modelling the objective is to compute the variation of state of observable physical quantities through a mesh representation of the physical domain. Thus, multidimensionality in scientific hypotheses defines a space where observations are made, differing from GIS applications in which time and space are objects characteristics.

Hypotheses modelling have been introduced in databases back in the 80's [3]. In that context, one envisioned a hypothetical database state, produced by delete and insert operations, and verified whether queries were satisfied on that hypothetical state. This approach is, however, far from the semantics needed in the experimental settings that we are interested in. Closer to our objective is the logical model proposed in the context of the HyBow project [15-16] for modelling hypotheses in the biology domain. The approach adopted by Hybrow supports hypotheses validation in the spirit

[1] `http://www.geneontology.org/`

of what we aim to represent, i.e., as a formal definition to be confronted with experimental results and that may extend the scientific knowledge base. However, the adopted ontological approach for hypothesis validation does not seem adequate for representing hypothesis-oriented research that considers quantitative validation of simulation results. In particular, the proposed approach in this paper aims to support the complete scientific exploration life cycle, and to the best of our knowledge, this is the first work that addresses this problem.

3 A Human Cardiovascular Simulation

In order to illustrate the issues involved in the *in-silico* modelling of natural phenomena, we refer to the hemodynamic based simulation of the human cardiovascular system developed at the LNCC, in support to medical diagnosis of heart diseases. Fig. 1 shows, in different scales, states of numerical simulations of the phenomenon, as described in [2].

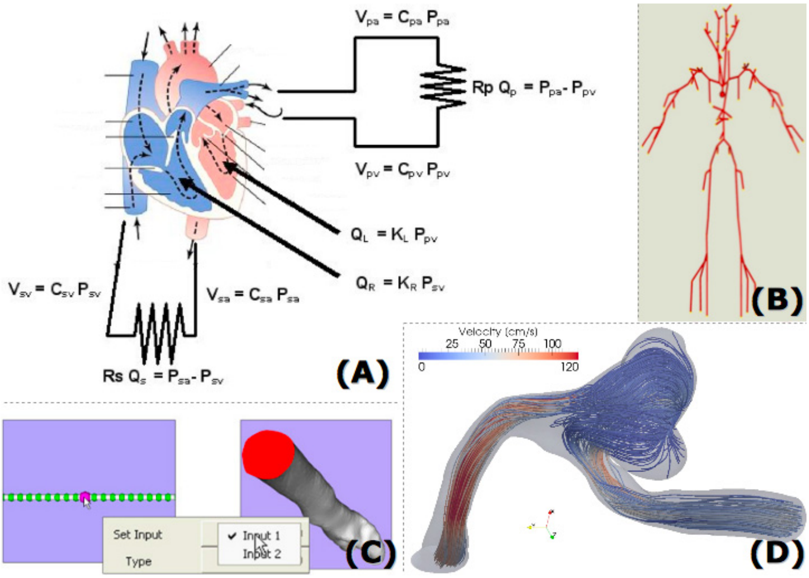


Fig. 1. Models of the Cardiovascular system in different scales

This example of scientific modeling activity starts with a simplistic representation of the human cardiovascular system, in which the parts of the system (a network of blood vessels) are modeled as lumped (non-spatial) physiological components [2]. These components, the blood vessels, are seen by analogy as resistive-capacitive electrical circuits, hence the same physical laws (e.g., Ohm's law) hold for them. This is the so-called 0-D model, which comprises, mathematically, ordinary differential equations. The cardiovascular system is then modeled as a lumped (closed-loop) dynamic system. Notice that we have here a hypothesis about how a generic component of the cardiovascular system behaves, viz., that (***h*$_1$**) *"A blood vessel behaves as a lumped RC-circuit"*.

In fact, the 0-D model is a simplistic representation of the cardiovascular system, yet (i) one which is the basis for more sophisticated models, and (ii) one that is able to achieve a number of relevant predictions; e.g., to predict how the patient's cardiac output and systemic pressure will change over his/her aging, or even the calibration itself of anatomical and physiological parameters by taking canonical values of an average individual.

Now, suppose that after a full research life-cycle, the distance between data produced by the simulation and data that has been observed is significant. It turns out that a model fine tuning (say, parameter recalibration) brings no sensible effect. The scientist might consider in this validation stage that hypothesis ***h*$_1$** does not seem to hold. In our example, feasible hypothesis reformulations could be (***h*$_2$**) *"A blood vessel behaves as a lumped RLC-circuit"* and *(**h$_3$**) "A blood vessel behaves as a lumped RC-circuit and an external pressure is exerted on it."*In (A) a 0D simulation computes physical quantities, pressure, flow and volume, independent of the space-time dimensions, modelling the heart chambers as a single point. In (B), a 1D simulation computes the cardiovascular system through a set of linear equations, fattening the space-time dimension. A more detailed simulation is found in (C). In this example, a usual strategy when modelling a complete system is to couple 3D simulations with simpler 1D models. The latter is input to the 3D model. Finally, (D) shows a 3D model based on a mesh structure that guides the space-time computation of physical quantities.

This example happens to present clearly the challenges involved in modelling the data computed by numerical simulations. First, a phenomenon must be observed through certain physical quantities claimed relevant for the model purpose. In the example of Fig. 1, those include: flow, velocity, pressure and displacement. The observation values establish the basis for model validation. Computation itself calculates values for observed physical quantities in points of a multidimensional grid, according to the chosen scale.

4 The Scientific Hypothesis Conceptual Model

In this section we present the scientific hypothesis conceptual model. The conceptual model extends in-silico experiments to cover the complete scientific exploration life-cycle from phenomenon observation and scientific hypothesis formulation to experimental validation and, eventually, hypothesis revaluation, see Fig. 2 .

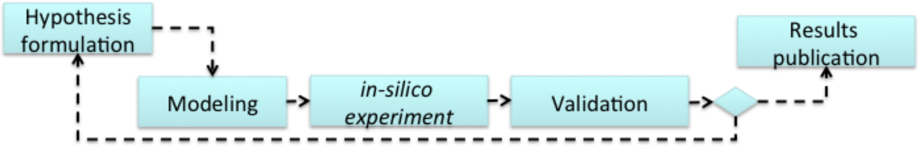


Fig. 2. In-Silico scientific exploration life-cycle

One of the applications of the proposed conceptual model is its implementation as a database comprising data and metadata about the scientific exploration life-cycle. The recordings therein obtained maybe used as: notebook of the scientific process; the source of provenance information [6]; the support for results reproducibility and scientific results analysis, just to name a few.

5 The Conceptual Model

Fig. 3 presents the scientific hypothesis conceptual model diagrammatic representation. It is structured around three main concepts: Phenomenon, Hypothesis and Phenomenon Process, Process for short.

6 The Phenomenon

The starting point of a scientific investigation comprehends the specification of the phenomenon, as the object of the research. In this conceptual model, a phenomenon is described as: **Ph** (Ph_{id}, Title, PQ, D) (1), where Ph_{id} is the phenomenon unique identifier; Title describes in natural language the observed phenomenon; PQ holds a set of physical quantities (i.e. pressure, velocity,…), whose values reflect the phenomenon state from the modeler perspective, and D is the dimensional space. The latter refers to the spatial scale to be considered S={0D,1D,3D} and the time frequency of observations.

7 The Scientific Hypothesis

A scientific hypothesis conceptually represents a formal model that provides a possible interpretation for the studied Phenomenon. Its conceptual representation enables expressing relationships between hypotheses, such as *evolves_from* and *is_composed_of*. The formalization of a scientific hypothesis is provided by a mathematical model, usually a set of differential equations, quantifying the variations of physical quantities in continuous space-time. At this point, the mathematical equations are represented in MathML, enabling models interchange and reuse.

As we are interested in modeling natural phenomena occurring in space-time, we borrow from Sowa's Process ontology [17] the specification of a scientific hypothesis from two perspectives, a continuous and a discrete process. The former refers to the mathematical model, earlier discussed, representing the studied phenomenon. The latter corresponds to the computational representation of the hypothesis that induces discrete transformations of the phenomenon state. In the cardiovascular simulation, the continuous process perspective is represented by a set of partial differential equations[2] that models the flow of blood through arteries and veins according to the

[2] Due to space restrictions, the differential equations are not presented. Interested readers are referred to [2] for a complete discussion on the topic.

specified scale (space, time), whereas the discrete process corresponds to the numerical simulation, including the mathematical solver (HEMOLAB) and the produced simulated data (see Section 5).

A given hypotheses may find its implementation in many computational models, using different numeric techniques, for instance. This is reflected in the cardinality of the represented_as relationship between Scientific Hypothesis and Discrete Process. It however is associated with a single continuous model. This is explained by the fact that the mathematical model is a formal and precise description of the hypothesis. Finally, different scientific hypotheses compete in trying to explain a natural Phenomenon.

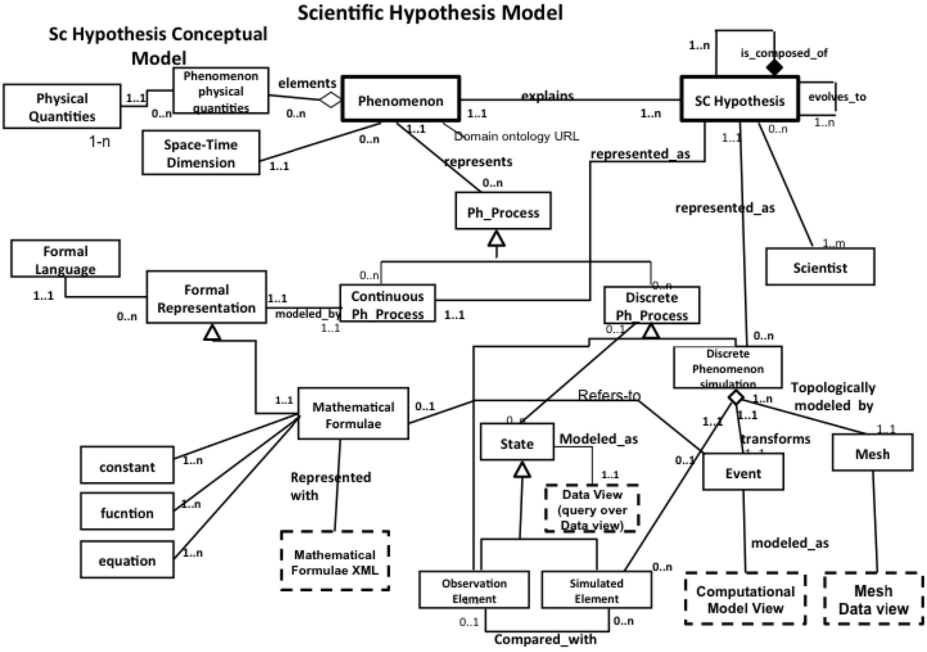


Fig. 3. Scientific Hypothesis Conceptual Model

In accordance with Sowa's Process ontology, the discrete process representation of scientific hypotheses is modeled by a composition of an *Event*, which applies transformations over the phenomenon state, and the data representation of the simulated states. Additionally, a *Mesh* corresponds to the topology of the physical domain in which the simulation takes place. In this conceptual model, a state reflects the values of the selected physical quantities in a space-time slice. Thus, an instance of discrete process unites the solver software that computes the simulation, the simulated data and the physical topology mesh.

Note that State entity generalizes both the Observation and Simulated data. Indeed, while simulated data is produced by discrete computation, observation data are also fruit of some kind of discrete collection. Finally, complementing the model description, the relationship *Refers_to* maps a mathematical model to its computational representation, instance of *Event.*

8 SciDB Implementation

In this preliminary experiment, we concentrate on using the SciDB multidimensional array structure [5] to store the states of scientific hypotheses computed by the simulations described in Section 3.

The multidimensional array structure is the basis for data representation in SciDB. A user specifies multidimensional structures by providing the range values for each dimension and a list of attribute values to compose each individual cell. In this context, the following mapping strategy has been defined: for each scientific hypothesis; i) define a multidimensional array object ii) specify the dimensions (D); iii) specify the list of physical quantities (PQ); iv) create an array having the dimensions as D and attributes as PQ. In this context, a 5 dimensional structure is used: *simulation* – enumerates each of the experiments; *t* is the time dimension, and <x,y,z> provides the 3D spatial coordinates. The Mesh physical implementation is not discussed. Using the AQL (Array Query Language), the following schema is defined to hold data from the 3D model of the artery in Fig. 1 (D).

```
CREATE ARRAY BloodFlow3D
    <velocity: point3D, pressure: double, displacement: point3D>
    [ simulations=0..*,0,0, t=0..500,500,0,
      x=1..7000,1000,0, y=1..7000,1000,0, z=1..36000,1000,0]
```

Observe that BloodFlow3D models the blood flow in D={3D,T_i}. The physical quantities listed as part of the array definition specify the values on each cell of the multidimensional grid, see Fig. 4. Thus, in fact the proposed logical model implemented in SciDB can be mapped directly to the conceptual model presented in Section 4, as far as the phenomenon state goes.

A data structure for representing the discrete process Event, in the form of a computational model, such as the Hemolab software, has been discussed elsewhere [12] and still needs to be integrated with the state strategy using SciDB.

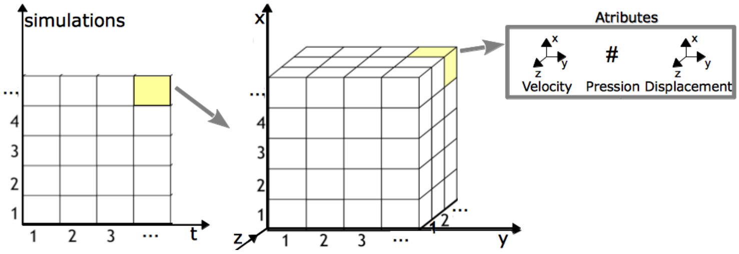


Fig. 4. Multiarray representation of the cardiovascular 3D simulation

9 Conclusion

Managing *in-silico* simulations has become a major challenge for eScience applications. As science increasingly depends on computational resources to aid solving extremely complex questions, it becomes paramount to offer scientists mechanisms to manage the wealth of knowledge produced during a scientific endeavor.

This paper presented a scientific hypothesis conceptual data model that aims to support the complete *in-silico* scientific exploration life-cycle. In extension to current practice, in which only the experimental phase is modeled using scientific workflows, we believe that in order to fully take advantage of *in-silico* resources, this practice has to be extended to enable the phenomenon description and the complete cycle of hypothesis validation. The proposed conceptual model follows this approach. It allows scientists to describe the observed phenomenon through elected physical quantities and to establish a multidimensional grid that guides physical quantities computation. The scientific hypothesis explains the corresponding phenomenon under a continuous and discrete process view.

The proposed conceptual model is the first, to the best of our knowledge, to offer an integrated view of the complete hypothesis based scientific exploration life-cycle. It can be used in support of the scientific life-cycle to: annotate the scientific process, register the formulated hypotheses; provide provenance information; support results reproducibility, validation and analysis.

As preliminary results, we use the SciDB system to store the results of a 3D simulation of a human cardiovascular system. We show that the basic principles used to model a hypothesis discrete state can be mapped into SciDB model, enabling its practical implementation using the system.

There are various opportunities for future work. We need to integrate the structures proposed in [12] to represent the discrete Event with the State as presented here. In particular, the descriptions of scientific workflows that compute the simulation states bring important information. The scientific hypothesis evolution and composition are important issues that shall enhance scientific life-cycle provenance. Finally, qualitative information regarding the domain in which phenomena are announced will complete the descriptive metadata discussed here.

References

1. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., Kepler, M.: An Extensible System for Design and Execution of Scientific Workflows. In: Proceedings of the SSDBM (2004)
2. Blanco, P.J., et al.: On the potentialities of 3d-1d coupled models in hemodynamics simulations. Journal of Biomechanics 42(7), 919–930 (2009)
3. Bonner, A.J.: Hypothetical Datalog: Complexity and Expressibility. Theoretical Computer Science 76, 3–51 (1990)
4. Clifford, B., et al.: Tracking Provenance in a Virtual Data Grid. Concurrency and Computation: Practice and Experience 20(5), 565–575 (2008)

5. Cudre-Mauroux, P., et al.: A Demonstration of SciDB: A Science-Oriented DBMS. In: 22th Int. Conference on Very Large Data Bases, Lyon, France (August 2009)
6. Freire, J., Koop, D., Santos, E., Silva, C.T.: Provenance for Computational Tasks: A Survey. Computing in Science and Engineering 10(3), 11–21 (2008), doi:10.1109/MCSE.2008.79
7. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studie 43(5-6), 907–928 (1995)
8. Mattoso, M., Werner, C., Travassos, G.H., Braganholo, V., Murta, L., Ogasawara, E., Oliveira, D., Cruz, S.M.S.D., Martinho, W.: Towards Supporting the Life Cycle of Large Scale Scientific Experiments. Int. J. Business Process Integration and Management 5(1), 79–92 (2010)
9. Oinn, T., Greenwood, M., Addis, M.: Taverna: Lessons in Creating a Workflow Environment for the Life Sciences. Concurrence Computation: Practice and Expeience, 1–7 (2000)
10. Parent, C., Spaccapietra, S., Zymanyi, E.: Conceptual Model for Traditional and Spatio-temporal Applications - The MADS approach. Springer (2006)
11. Porto, F., de Macedo, J.A., Tamargo, J.S., Zufferey, Y.W., Vidal, V.P., Spaccapietra, S.: Towards a Scientific Model Management System. In: Song, I.-Y., Piattini, M., Chen, Y.-P.P., Hartmann, S., Grandi, F., Trujillo, J., Opdahl, A.L., Ferri, F., Grifoni, P., Caschera, M.C., Rolland, C., Woo, C., Salinesi, C., Zimányi, E., Claramunt, C., Frasincar, F., Houben, G.-J., Thiran, P. (eds.) ER Workshops 2008. LNCS, vol. 5232, pp. 55–65. Springer, Heidelberg (2008)
12. Porto, F., Moura, A.M.C.: Scientific hypothesis database. Technical Report, LNCC (2012)
13. Porto, F., Tajmouati, O., Silva, V.F.V., Schulze, B., Ayres, F.M.: QEF Supporting Complex Query Applications. In: 7th Int. Symposium on Cluster Computing and the Grid, Rio de Ja-neiro, Brazil, pp. 846–851 (2007)
14. Racunas, S.A., Shah, N.H., Albert, I., Fedoroff, N.V.: Hybrow: a Prototype System for Computer-Aided Hypothesis Evaluation. Bioinformatics 20(suppl.1), 257–264 (2004)
15. Racunas, S., Griffin, C., Shah, N.: A Finite Model Theory for Biological Hypotheses. In: Proc. of the 2004 IEEE Computational Sytems Bioinformatics Conference (2004)
16. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foun-da-tions, Brooks Cole Publishing Co., Pacific Grove, CA, ©2000 (August 1999)
17. Stonebreaker, M., Rowe, L.A.: The Design of Postgres. In: SIGMOD Conference, pp. 340–355 (1986)
18. Stonebreaker, M., Becla, J., DeWitt, D., et al.: Requirements for Science Data Base and SciDB. In: Conference on Innovative Data Systems Research, CIDR (2009)

An Integrated Multidimensional Modeling Approach to Access Big Data in Business Intelligence Platforms

Alejandro Maté, Hector Llorens, and Elisa de Gregorio

Lucentia Research Group, Department of Software and Computing Systems,
University of Alicante, Spain
{amate,hllorens,edg12}@dlsi.ua.es

Abstract. The huge amount of information available and its heterogeneity has surpassed the capacity of current data management technologies. Dealing with that huge amounts of structured and unstructured data, often referred as Big Data, is a hot research topic and a technological challenge. In this paper, we present an approach aimed to allow OLAP queries over different, heterogeneous, data sources. The modeling approach proposed is based on a MapReduce paradigm, which integrates different formats into the recent RDF Data Cube format. The benefits of our approach are that it allows a user to make queries that need data from different sources while maintaining, at the same time, an integrated, comprehensive view of the data available. The paper discusses the advantages and disadvantages, as well as the implementation challenges that such approach presents. Furthermore, the approach is illustrated in an example of application.

Keywords: Conceptual models, Business Intelligence, Big Data.

1 Introduction

Nowadays, the amount of information available on the Internet can go up to hundreds of petabytes or even exabytes. It is already not possible to process, store, and manage all this information in local servers even for the biggest companies Business Intelligence (BI) systems. The possibility of making on-line analytical processing (OLAP) queries over high amounts of information, while being able to retrieve only the relevant information at each moment, would provide important benefits. However, given the heterogeneity and size of the data used, the effort required to harness the power of all this information can not be afforded by individual companies. This phenomena is referred to as Big Data [3].

Until the recent years, not much structured information was available on the Internet. Most of this information was textual information written in the most widely spoken natural languages. Structured data was principally stored in private databases, being accessible only by their owners. Nowadays, another breaking phenomena, Open-Data, is changing this situation drastically. In the same

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 111–120, 2012.

manner as Wikipedia freely brings unlimited access to lots of semi-structured information, many institutions and communities have decided to publish and share on the Internet the information they manage. For example, the governments of some countries have decided to publish their information in order to increase their transparency (e.g. data.gov.com). Following this trend, many other kinds of data, such as road traffic, are also becoming open, thus increasing the number of available sources of data.

In this context, imagine the possibility of performing OLAP analysis over a distributed model mixing private local data with the open linked data available. A company with a private database (sales, customers, strategy) could benefit from querying this data together with the linked data, which can provide a significant enhancement for the company OLAP capabilities. The following are just few examples of what could be done with such a model: "Which countries are suffering decreases in sales and GDP drops" (WorldBank stats), "Which of our products decreasing in sales in the last quarter have negative opinions on Twitter?"

This kind of queries would clearly provide a good support to improve the decision making process in companies. The difficulty behind this challenge is mainly focused on the heterogeneity of the sources from which the information would be extracted, as well as on the importance of an efficient distributed model to make these queries computationally viable.

The aim of this paper is to propose and analyze an integrated approach to allow OLAP queries over heterogeneous data sources, where each data source may contain different internal and external dimensions. We propose an approach based on the MapReduce strategy capable of dividing a query, and distributing it to different nodes that access different datasets in a variety of formats. The output of these nodes is then seamlessly integrated, making the process transparent for the user. In order to increase the extensibility of our proposal, we base our approach in standards, using SPARQL as the query language for the distributing/integrating module and as input for the nodes.

The remainder of this paper is structured as follows. Related work is reviewed in Section 2. Our proposal of an integrated model to access Big Data is presented in Section 3. Finally, in Section 4 we include a discussion of the advantages of the model and the difficulties related to its implementation, as well as the further directions of this research.

2 Related Work

In this section, we present the different technologies related to our proposal. First, Big Data and distributed architectures are reviewed. Then, we analyze Linked Data, RDF and SPARQL standards. Afterwards, we discuss the current Data warehouse (DW) and MD modeling proposals. Finally, the contribution of the paper w.r.t. the current state-of-the-art is summarized.

2.1 Big Data and Distributed Architectures

The current efforts to manage Big Data are centered in distributed architectures, among which MapReduce [13] is one of the most commonly applied. Some relevant implementations of this approach are Hadoop[1], Hive[2], MongoDB [3].

MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes). Basically, the framework presents a node which performs the task of distributing pieces of information to other nodes in the network. Then, it applies a reduce function on the result retrieved. In this approach, each node is responsible for obtaining a partial result of the process.

2.2 Linked Data, RDF and SPARQL

Linked data is based mainly on two standards: RDF and SPARQL. On the one hand, the RDF model [4] encodes data in the form of triples (subject, predicate, object). These three elements are represented by Uniform Resource Identifiers (URIs) [7] with the exception of object, which can also be represented by a literal i.e., string, number, etc. In short, this allows us to assert resource A has some relation with resource or literal B. In this way, we can also relate resources from different sources making links between them. RDF is flexible to the point that vocabularies and ontologies can be created by users using standard vocabulary definition languages (RDFS) or ontological languages (OWL). Recently, a new feature relevant for this paper was added to SPARQL, federated queries. The new SERVICE keyword extends SPARQL to support queries that request data distributed across the Web in different end-point nodes. Finally, focusing on extending OLAP capabilities of companies, there exist RDF vocabularies [4], aimed to support multidimensional data and statistical data.

2.3 OLAP, Data Warehouses and Multidimensional Data

Traditional MD modeling approaches [1,6,11] focus on modeling the target DW, which will serve as basis for the BI platform. In this way, elements modeled are either (i) part of the underlying DW, or (ii) part of the data sources, from where the data will be extracted. Therefore, these approaches assume that, all the data which may possibly be queried, is being stored (i) under the same schema (ii) in an integrated manner. However, when we consider situations where big amounts of heterogeneous data are present [9], it is possible that (i) we do not require all the information at the same time, and (ii) data is stored according to different schemas and technologies in different places. Since traditional approaches do not provide a mechanism to specify which data should be retrieved from each schema, and how the integration should be performed, we require the addition

[1] http://hadoop.apache.org/

[2] http://hive.apache.org/

[3] http://www.mongodb.org/

[4] http://www.w3.org/2011/gld/wiki/Data_Cube_Vocabulary

of new constructs in order to capture this information while, at the same time, preserving the rich semantics of specific technologies such as RDFs.

2.4 Similar Proposals and Our Contribution

Previously, other proposals have focused on distributing SPARQL queries. In [10], a method to query multiple SPARQL end-points in an integrated way is presented. At that time (2008), SPARQL did not support federated queries natively, thus different proposals extending it with this capability were presented. However, nowadays, this functionality is already included in the SPARQL definition. In our paper, in addition to benefit from distributed queries, we present an architecture not only capable of integrating RDF/SPARQL data sources, but also other types such as Mondrian Warehouses and Web APIs.

In [8], a general and efficient MapReduce algorithm for SPARQL Basic Graph Pattern is presented. This improves the performance and scalability of join operations. However, this approach does not consider the possibility of including different types of nodes into the architecture.

In [5], the authors discuss the implementation of a distributed SPARQL query engine over a Hadoop cluster. They optimize the SPARQL performance in one individual server containing RDF graphs using the MapReduce strategy. Unlike this proposal, our aim is to offer access to different endpoints. Therefore, our work could be considered complementary, focusing on a more general level. If the SPARQL end-points in our approach are optimized using this technique, the overall result would probably also be optimized.

In a nutshell, there is an absence of general integration proposals such as the one being proposed in this paper. The MapReduce strategy has been applied over individual SPARQL graphs, but not over a grid of heterogeneous data source nodes. According to [2], "...the key benefit of Linked Data from the user perspective is the provision of integrated access to data from a wide range of distributed and heterogeneous data sources". Thanks to our approach, users have the possibility of querying their own private data together with the public open-data offered in either SPARQL, by means of an API or other in any other fashion. Furthermore, our proposal is focused on a multidimensional model rather than a simple integrated model as proposed by related works.

3 An Integrated Model to Access Big Data

In this section we describe our proposal to provide users with access to Big Data structures in a seamlessly way while preserving the structure of the information.

As mentioned in Section 1, Big Data is conformed by huge amounts of heterogeneous information from different sources. These huge amounts of data present two main problems to be included in a traditional DW structure: (i) the high cost of querying and maintaining the information [9], and (ii) some of the semantic relationships cannot be represented in traditional MD schemata [1,6], such as concept relationships in ontologies [12]. Therefore, as these relationships cannot be captured, valuable information and analysis capabilities are lost.

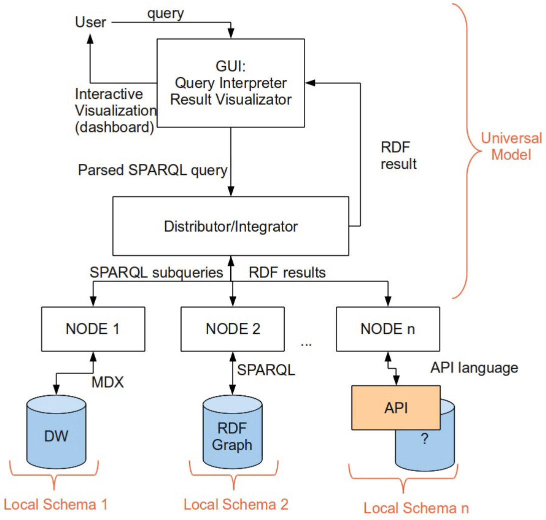


Fig. 1. Overview of the proposed architecture

In order to overcome these issues, our proposal distributes the multidimensional schema and the responsibility for providing information among nodes which compose the network, as can be seen in Figure 1. Each node is responsible for interpreting the queries it receives and retrieving its own data. Finally, the node sends the requested data back to the central module, which integrates these data according to the Universal Schema, using the MapReduce paradigm. Once the result of the query is integrated in the Universal Schema, it is presented to the user as in a traditional approach.

As opposed to traditional MD modeling, where the whole structure of the cube is known before-hand, our proposal allows the user to query certain information which does not appear in the Universal Schema. This is done by means of query resolution delegation, where certain parts of the query are fully resolved only by an specific node, and the results are, afterwards, integrated into the Universal MD Schema presented to the user. As such, our modeling proposal, exemplified in Figure 2, considers standard dimensions, which present a set of attributes along with a well-defined hierarchy (stockmarket, time, geolocation, etc.), as well as external elements (marked in the figure with *), which act as partially defined dimensions, levels or additional information with which to enrich OLAP analysis.

External elements are captured in a two-step process. First, an extension of traditional MD modeling is performed. In this extension, we add a set of new constructs which consider the possibility of designing MD elements classified as external. Then, these elements are assigned to their corresponding nodes by means of a deployment diagram and a Local Schema is created, representing, at least, the common attributes required for the query join.

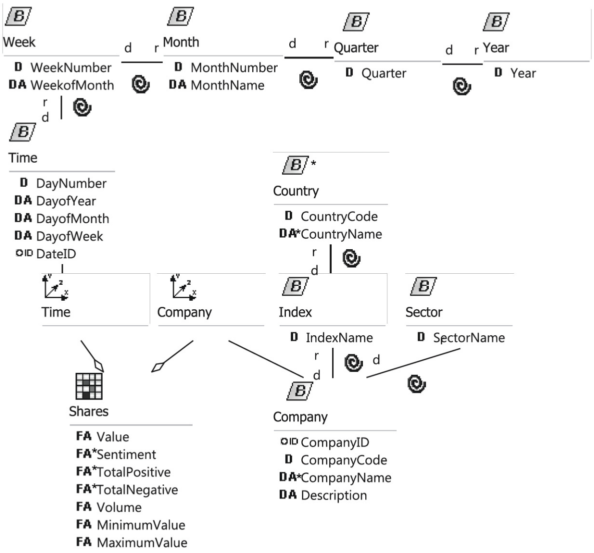


Fig. 2. A Universal Schema modeling Shares analysis

3.1 Multidimensional Model Extension

Traditionally, the main elements involved in MD modeling [1,6] are: (i) Facts, e.g. Shares, which are the center of analysis, (ii) Dimensions, e.g. Company, which represent the context of analysis, (iii) Bases, e.g. Index, inside the Company dimension, which constitute levels of aggregation in a given dimension, (iv) Descriptors, e.g. CompanyCode, which constitute attributes that differentiate instances in the same Base level, (v) Dimension Attributes, e.g. CompanyName, which provide additional descriptive information, and (vi) Fact Attributes (Measures), e.g. Value, which provide information about the performance of the process being analyzed. Among these elements, any of them can be considered external, with the sole exception of Facts, since they define which Dimensions and Fact Attributes are involved in the analysis, as well as the granularity of the tuple. A summary is depicted in Figure 3.

When modeling external elements, some special considerations are included in addition to the data being stored in a remote location:

1. Every external element modeled in the Universal Schema must appear in the corresponding Local Schema of the Specialized Node. This condition guarantees that the Universal Schema is consistent with the information contained in the Specialized Nodes.
2. An External Dimension is a Dimension which is completely delegated to a Specialized Node. These dimensions are characterized by presenting only the lowest Base level of the hierarchy in the Universal Schema, which is also marked as external.

3. An External Base is a Base level which specifies only the Descriptor of the Base level and the necessary Dimension Attributes for integrating the data retrieved from the Specialized Node. Therefore, these Dimension Attributes present in the Universal Schema, must appear in the Local Schema as Descriptors of the Base level.
4. An External Dimension Attribute is a Dimension Attribute whose value is retrieved from a Specialized Node for, at least, one or more instances of the corresponding Base level.
5. External Fact Attributes are Fact Attributes whose value is retrieved from a Specialized Node. Since Facts are analyzed according to a set of Dimensions, in order to provide the correct value for the corresponding tuple, the Specialized Node which contains the External Fact Attribute must be able to obtain the necessary information to integrate each value retrieved with the corresponding fact tuple. In turn, this means that the Specialized Node must be aware of all the dimensions involved in the analysis.
6. Unlike External Dimension Attributes, External Fact Attributes are usually linked to a certain Date. For example, share values vary continuously, thus each value is linked to a certain point in Time. However, information obtained from external sources does not always include historical information, thus we can only retrieve its current value. As such, we differentiate between two different kinds of External Fact Attributes: Proper and Improper. A Proper External Fact Attribute is an External Fact Attribute which, in addition to being linked to its corresponding instance on each non-Time dimension, it is also linked to the corresponding Date. On the other hand, an Improper External Fact Attribute is an External Fact Attribute which is not related to a certain Date, thus it only presents its current value.

These external elements are modeled in the same way as traditional MD elements with the exception of the previous considerations and the corresponding semantic differences. Therefore, the simplest way to allow the modeling of external elements is to consider adding a property to the abstract class MDElement indicating if the element is external or not, and restricting this value in the case of Facts.

3.2 Deployment Diagram

In order to preserve the information about which information is provided by each node, a deployment diagram similar to UML deploy diagrams is modeled, considering MD elements as components which can be deployed into the different nodes participating in the MD schema. This is shown in Figure 4.

After modeling the external elements in the Universal Schema, a Local Schema is created for each node which, when combined with the Universal Schema, determines the way data is integrated (joined) in the queries. In order to guarantee the correct integration of data retrieved from each node, each Local Schema must present every element assigned to its Node in the deployment diagram, although the type of each element may vary in the case of Dimension Attributes and Descriptors, as they act as union points between the sets of data retrieved.

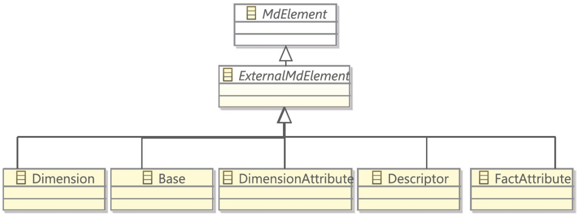


Fig. 3. External MD element list without considering their relationships

If we consider the architecture depicted in Figure 1, the models presented in Figure 2 and Figure 5, and the query (for the sake of clarity written in natural language) "Information about Companies from Countries with GDP growth greater than 1% and having more than 200 opinions in Twitter, including their difference in shares Value, and their Country", the query resolution process would be performed as shown in Figure 6.

First, the list of companies would be retrieved from the local DW. Then, for each company obtained, the Twitter node would retrieve which ones have more than 200 opinions. This information would be joined through CompanyName, as specified by the Universal and Local schemata. Simultaneously, the RDF node would retrieve the GDP growth for each country and return which countries present a grow rate bigger than 1%. In this case, the result would be joined through CountryName, finally obtaining the answer to the initial query.

Do note that, as shown in this example, as query resolution is delegated to the different nodes, the user can include a property in the query (GDP growth) whose relationship with the level (Country) has not been modeled in the Universal Schema. However, since the RDF node is able to identify and locate this property (and others it may store), the requested result is retrieved and sent back to the integrator node, who integrates the data from the different nodes and provides the user with the results of the query. In this way, each node can maintain its own representation of the data stored, without restricting this representation to MD models.

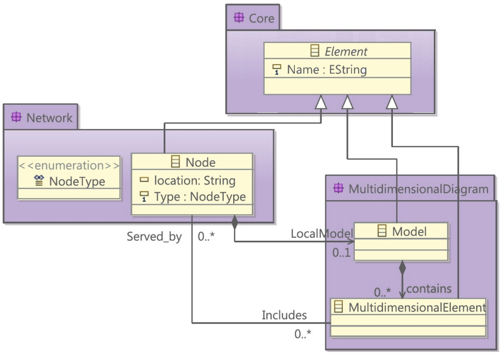


Fig. 4. (a) Deployment metamodel

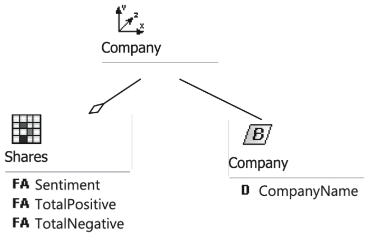


Fig. 5. (b) Local schema

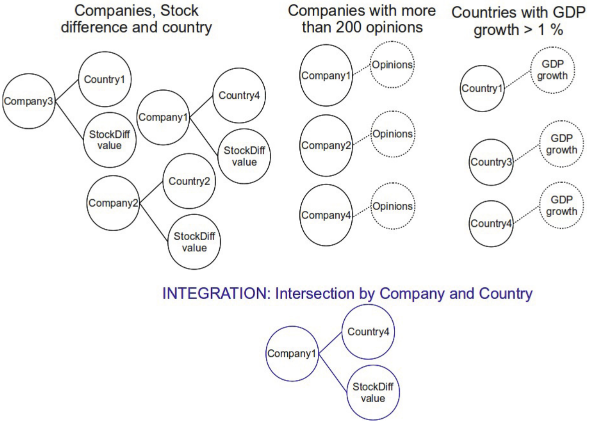


Fig. 6. Query resolution process in our proposal

4 Discussion and Future Work

In this paper we have presented an approach to allow querying Big Data structures, divided into several nodes, in an integrated way. In our approach, each node providing information maintains its internal structure, thus there is no information loss. The main advantages of our approach are that (i) it provides a unified vision of the data, allowing to add and remove nodes and information in a seamlessly way, and (ii) maintains the structure of the information on each node, thus rich semantic relationships are preserved and can be queried even if this information is not present in the universal schema.

Parallelizing the queries presents some advantages and disadvantages in contrast to sequential distribution. On the one hand, if queries are made sequentially, each subsequent query is more limited and produces less results. However, the waiting time of the sequential querying strategy and filtering requirements of each subsequent query slow down the process. On the other hand, if queries are made in parallel, the results produced in the different nodes are greater in size and the main filtering is produced in the integration step. The queries are faster but the data to be transmitted and integrated afterwards is bigger. This strategy has two main advantages if a query involves many nodes: (i) there is no need to wait for node responses before sending all the queries, and (ii) the integration can be done gradually as the responses are received by the integrator.

The main future work is to carry out an efficient implementation of the model and evaluate it. Optimizing the query process can result into a powerful querying tool to integrate local data with extra information coming from the web, thus enriching current OLAP analysis. In order to test the applicability of our approach, the model will be tested in real companies to which their data will remain private but with the presented model will be linked to a set of public open-data sources, thus improving the company OLAP capacity.

Finally, we consider improving the interface with a natural language processing module acting as a question answering system in which the queries can be introduced in a controlled English language, thus making it simpler to query the data and avoiding to specify complex SPARQL queries.

References

1. Abelló, A., Samos, J., Saltor, F.: Yam2: a multidimensional conceptual model extending uml. Information Systems 31(6), 541–567 (2006)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) 5(3), 1–22 (2009)
3. Cohen, J., Dolan, B., Dunlap, M., Hellerstein, J., Welton, C.: Mad skills: new analysis practices for big data. Proceedings of the VLDB Endowment 2(2), 1481–1492 (2009)
4. Klyne, G., Carroll, J., McBride, B.: Resource description framework (rdf): Concepts and abstract syntax. W3C recommendation 10 (2004)
5. Kulkarni, P.: Distributed SPARQL query engine using MapReduce. Master's thesis, http://www.inf.ed.ac.uk/publications/thesis/online/IM100832.pdf
6. Luján-Mora, S., Trujillo, J., Song, I.: A uml profile for multidimensional modeling in data warehouses. Data & Knowledge Engineering 59(3), 725–769 (2006)
7. Masinter, L., Berners-Lee, T., Fielding, R.: Uniform resource identifier (uri): Generic syntax (2005)
8. Myung, J., Yeon, J., Lee, S.G.: Sparql basic graph pattern processing with iterative mapreduce. In: Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud, MDAC 2010, pp. 6:1–6:6. ACM (2010)
9. Niemi, T., Niinimäki, M., Nummenmaa, J., Thanisch, P.: Constructing an olap cube from distributed xml data. In: Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP, pp. 22–27. ACM (2002)
10. Quilitz, B., Leser, U.: Querying Distributed RDF Data Sources with SPARQL. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 524–538. Springer, Heidelberg (2008)
11. Tryfona, N., Busborg, F., Borch Christiansen, J.: starer: A conceptual model for data warehouse design. In: Proceedings of the 2nd ACM International Workshop on Data Warehousing and OLAP, pp. 3–8. ACM (1999)
12. Uschold, M., Gruninger, M.: Ontologies: Principles, methods and applications. Knowledge Engineering Review 11(2), 93–136 (1996)
13. White, T.: Hadoop: The Definitive guide (2009)

A Comprehensive Model for Provenance

Salmin Sultana and Elisa Bertino

Purdue University
{ssultana,bertino}@purdue.edu

Abstract. In this paper, we propose a provenance model able to represent the provenance of any data object captured at any abstraction layer and present an abstract schema of the model. The expressive nature of the model enables a wide range of provenance queries. We also illustrate the utility of our model in real world data processing systems.

Keywords: Provenance, Accountability, Security, Workflow Systems.

1 Introduction

Existing data provenance systems mostly operate at a single level of abstraction at which they record and store provenance. Provenance systems for scientific data [1][2] record provenance at the semantic level of the application. Other application level provenance systems capture provenance at the granularity of business objects, lines of source code or other units with semantic meaning to the context. Workflow systems record provenance at workflow stages and data/message exchange points. System-call based systems [3][4] operate at the level of system processes and files. While provenance collected at each abstraction layer is useful in its own right, integration across these layers is crucial.

To build a unified provenance infrastructure, defining an expressive provenance model able to represent the provenance of data objects with various semantics and granularity is the first crucial step. Such a model should be able to capture data provenance in a structured way as well as to encapsulate the knowledge of both the application semantics and the system. The model should also support provenance queries that span layers of abstraction. Despite a large number of research efforts on provenance management, only a few provenance models have been proposed. Moreover, most of these models are specific to a provenance system and conform only to that particular system's data structure. Although a general provenance model has been proposed by Ni et al. [5], its main focus is on access control for provenance. Also this model is not able to distinguish between application and system level provenance.

We have proposed a provenance model [6] that is (i) generic enough to record the provenance of any data object, (ii) unified to capture and integrate both the application and system level metadata, and (iii) tailored to fine grained access control and originator preferences on provenance. In this paper, we analyze the requirements that any comprehensive provenance model should satisfy, present the model, and then illustrate the utility of our model in real world data processing systems.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 121–130, 2012.

2 Requirements

In order to provide a generic provenance structure for all kinds of data objects, the provenance model must meet the following requirements:

Unified Framework: The model must be able to represent metadata provided by the various provenance systems. Although a number of system-call based provenance architectures [3] [4] have been proposed to capture file provenance, there is no well defined model to represent and organize such low level metadata. One important goal for any comprehensive provenance model is to bridge this gap and provide a unified model able to represent provenance for any kind of data at any abstraction layer. To this end, it is crucial to identify a comprehensive set of features that can characterize the existing provenance systems and systemize provenance management.

Provenance Granularity: Provenance may be fine-grained, e.g. provenance of data tuples in a database [7], or coarse-grained, such as for a file in a provenance-aware file system [4]. The usefulness of provenance in a certain domain is highly related to the granularity at which it is recorded. Thus, the provenance model should be flexible enough to encapsulate various *subjects and details of provenance* based on user specifications.

Security: The model must support provenance security. Access control and privacy protection are primary issues in provenance security. To meet these requirements, the provenance model must support the specification of privacy-aware fine grained access control policies and user preferences.

Interoperability: A data object can be modified by and shared among multiple computing systems. Hence, the provenance model must support provenance interoperability i.e. the integration of provenance across different systems.

Provenance Queries and Views: The model should support various types of provenance queries. Historical dependencies as well as subsequent usages of a data object should be tracked easily. If a data is processed in multiple system domains, an administrator might want to see a high level machine, system or domain view of the provenance graph. In addition, to find relevant information from large provenance graphs, one should be able to filter, group or summarize all/portions of provenance graphs and to generate tailored provenance views. Thus, the model should be able to distinguish the provenance generated from different systems and construct **specialized views** of provenance graphs.

3 Provenance Model

Fig. 1 shows the proposed provenance model consisting of entities and the interactions among them. To characterize our model, we define the provenance as:

Definition (Provenance). *The provenance of a data object is the documented history of the actors, process, operations, inter-process/operation communications, environment, access control and other user preferences related to the creation and modification of the object. The relationships between provenance entities form a provenance graph (DAG) for the data object.*

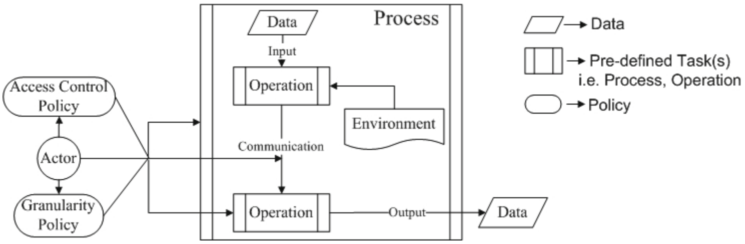


Fig. 1. Proposed Provenance Model

Data creation or manipulation is performed by a sequence of *operation*s initiated by a *process*. A *process*, consisting of a sequence of operations, may be a service/activity in a workflow, a user application, or an OS-level (e.g. UNIX) process. An *operation* executes specific task(s) and causes manipulation to some system or user data. Thus, the operations do not only generate/modify persistent data but also generate intermediate results or modify system configurations. *Communication* represents the interaction (e.g. data flow) between two processes or two operations in a process. Communication between two operations in a process means the completion of an operation following the start of another operation. When the preceding operation results in data, the communication may involve data passing between the operations. The communication may also contain triggers, specific messages, etc. However, in most of the cases there might be no explicit message (i.e. communication record) exchange between two operations. Web service, user application, and UNIX process are examples of *process*es; statements within an executable, function, command line, etc. exemplify the *operation*s; while data flow, copy-paste, inter-process communication in UNIX, etc. represent the *communication* between operations or processes.

An operation may take data as input and output some data. Each data object is associated with a *lineage* record which specifies the immediate data objects that have been used to generate this data. *Lineage* is particularly helpful for producing the data dependency graph of a data object.

Processes, operations, and communications are operated by *actor*s that can be human users, workflow templates, etc. Where data provenance is used to detect intrusion or system changes, the knowledge of a user role or the workflow template may be helpful. *Environment* refers to the operational state, parameters, system configurations that also affect the execution of an operation and thus output data. This additional provenance information is crucial for understanding the performance of the operation and the nature of the output.

Security and privacy of provenance are crucial since data or provenance may contain sensitive or commercially valuable information. The nature of this confidential information is specific to the applications and hence the protection policies and the access control can be handled by the involved actors. To address these requirements, *access control policies* by actors are included in the provenance model. These access control policies specify whether and how other actors may utilize process, operation, communication and lineage records.

Since our provenance model can capture the very details of an operation, it might by preferable to allow users to specify the desired level of provenance details. For example, in a scientific workflow, it may suffice to capture the provenance information in a service/activity whereas in a command line (e.g. *sort*), it may be required to record the OS level operations, system configuration etc. The *granularity policies* allow the users to specify how detailed provenance information they want to be captured and stored.

Table 1. Mapping between the entities in OPM and our model

Property	OPM Entity	Entity in our Model
Physical or digital data object	Artifact	Data Object
Action(s) performed on or by artifacts	Process	Process consisting of Operations and Communications
Contextual entity controlling process execution	Agent	Actor, Environment

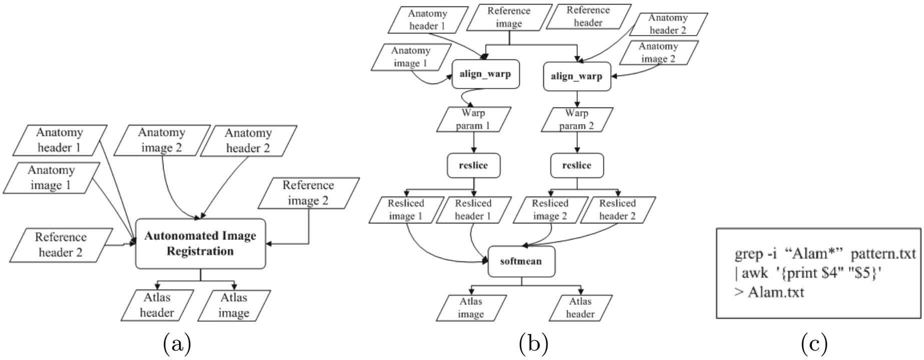
Our model conforms to the OPM representation. Provenance in OPM is described using a directed graph consisting of entities with connecting edges [8]. OPM entities are of three types, namely *artifact*, *process*, *agent*. There are five types of edges which represent the causal dependencies amongst entities. Table 1 shows how our provenance model complies with the OPM by listing the OPM entities and their counterparts in our model.

3.1 Use Case

To illustrate our provenance model, we consider some use cases and identify the provenance entities in these contexts.

Figure 2(a) shows a workflow example from the field of functional MRI research [9], where brain images of some subjects are spatially aligned and then averaged to produce a single image. The workflow contains the automated image registration (AIR) process that operates on a collection of anatomy images and produces an averaged brain image. An *actor* (e.g. an administrator of the experiment system) specifies a *granularity policy* for automated provenance collection to capture provenance at the process granularity. In this context, the provenance for 'Atlas image' and 'Atlas header' contains the AIR *process* with anatomy and references images & headers as the input *lineage* data. Since no details about the AIR process are captured, we assume the process consists of a single *operation* named as AIR. Figure 2(b) presents the breakdown of the AIR process into operations and interactions between them. If a user defined policy requires to capture operation level provenance, the provenance graph for 'Atlas image' will contain the AIR *process* with operation hierarchy **align_warp** -> **reslice** -> **softmean**. The data flow between operations represents their communication; for example the transfer of **Warp param 1** indicates the communication between **align_warp** and **reslice** operations. However, the data dependency graph of 'Atlas image' contains the input images as well as all the intermediate results.

Finally, we consider a UNIX shell script - ‘pattern.sh’, shown in Figure 2(c) to show the applicability of our model to provenance aware file/storage systems, operating systems, etc. The script uses the ‘grep’ command to extract all the patterns starting with ‘Alam’ from the ‘data.txt’ file and sends the output to the ‘awk’ command through a pipe. The ‘awk’ command then extracts particular information from the input data and writes the information in the output file ‘Alam.txt’. The execution of the script (namely ‘pattern’ *process*) may be assigned a unique process ID by the system. The process consists of two operations, ‘grep’ and ‘awk’. Thus, the provenance of ‘Alam.txt’ contains the operation dependency **grep -> awk** and the data dependency on ’data.txt’ and the intermediate pipe (uniquely identified by an ID).



```
grep -i “Alam*” pattern.txt
| awk '{print $4" "$5}'
> Alam.txt
```

(a) (b) (c)

Fig. 2. (a) Workflow for ‘Automated Image Registration’ (AIR) process operating on a series of images & headers and producing an average image according to different axes. (b) Break down of AIR process into operations and data flows between them. (c) A shell script representing a user program and corresponding OS-level process.

3.2 Provenance Records

Data provenance is stored as a set of provenance records in a *provenance repository* [5]. Provenance storage, manipulation and query can be implemented using data management systems characterized by different data models such as the relation model, XML, and RDF. Since our provenance model is generic, we do not specify implementation details here. We represent our model as the relationships among the following provenance records (see Fig 3): (i) *Process* (ii) *Operation* (iii) *Communication* (iv) *Actor* (v) *Environment* (vi) *Lineage* (vii) *Access Control Policy* (viii) *Granularity Policy*. Each data object and provenance record is uniquely identified by an ID attribute. Since provenance information may be exchanged across different systems, we use *domain* to specify the system where the executions and data manipulations occur. The *domain* value may include a particular application, a workflow, a machine, a system domain, or any combination of these. This attribute is extremely useful when customizing the provenance graph to efficiently generate an **abstract domain view**.

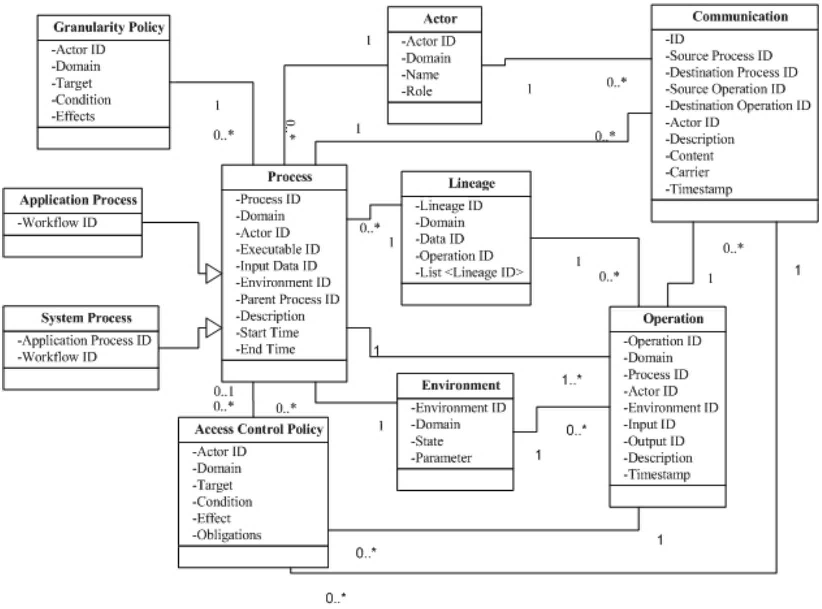


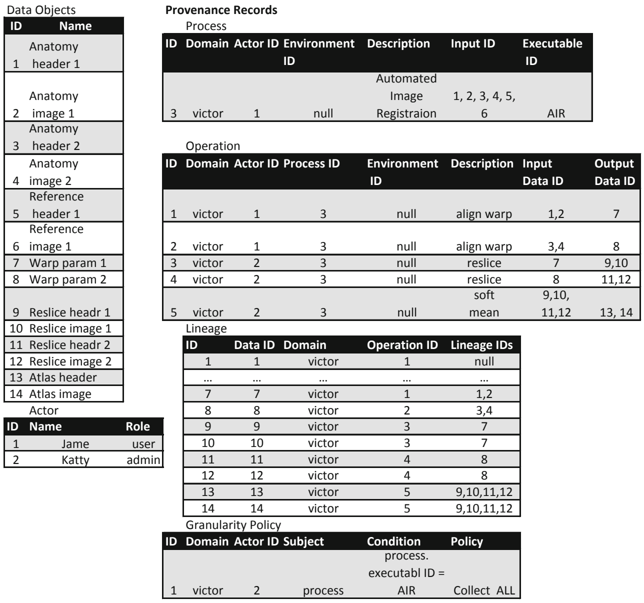
Fig. 3. Class Diagram of Provenance Model

We describe a process with the base class *process* and differentiate between the high level and the system process by creating two inherited classes of *process*. Each *process* is executed by an *actor* in a certain computational *environment* and may generate *output* data. If the *process* is part of a scientific workflow, web service, etc., it is distinguished by the subclass *Application Process* which also contains the workflow ID. The *System Process* class describes the OS level processes and possesses workflow ID as well as the host application process ID.

Depending on the applications, the description of an *Operation* or *Communication* may contain a statement or a block of statements, a function defined by pseudo-code or source code, but it can also be only a function name. The carrier of a *communication* includes the message transferring channel, e.g. email, which may be sensitive and useful in some cases, e.g. digital forensics.

Access Control Policy record attributes include policy ID, actor ID, subject, condition, effect, obligations. The actor ID logs the author of the record. The subject attribute is used to specify the record(s) at which the access control aims. The subject of an access control policy record only refers to a process, operation or a communication record. *Granularity Policy* record comprises of policy ID, actor ID, subject, condition and policy attributes. An actor may define policies to capture provenance only at the process level or to exclude the lineage information for a particular application. Subject states the targeted record at which the granularity policy applies based on the condition value.

To illustrate the application of provenance records to the use cases from section 3.1, we consider a RDBMS implementation of the provenance storage. Figure 4 shows the data objects and related provenance records generated from the workflows in 2(b). For simplicity, we do not show some attributes.



Data Objects

ID	Name
1	Anatomy header 1
2	Anatomy image 1
3	Anatomy header 2
4	Anatomy image 2
5	Reference header 1
6	Reference image 1
7	Warp param 1
8	Warp param 2
9	Reslice headr 1
10	Reslice image 1
11	Reslice headr 2
12	Reslice image 2
13	Atlas header
14	Atlas image

Actor

ID	Name	Role
1	Jame	user
2	Katty	admin

Provenance Records

Process

ID	Domain	Actor ID	Environment ID	Description	Input ID	Executable ID
3	victor	1	null	Automated Image Registraion	1, 2, 3, 4, 5, 6	AIR

Operation

ID	Domain	Actor ID	Process ID	Environment ID	Description	Input Data ID	Output Data ID
1	victor	1	3	null	align warp	1,2	7
2	victor	1	3	null	align warp	3,4	8
3	victor	2	3	null	reslice	7	9,10
4	victor	2	3	null	reslice	8	11,12
5	victor	2	3	null	soft mean	9,10, 11,12	13, 14

Lineage

ID	Data ID	Domain	Operation ID	Lineage IDs
1	1	victor	1	null
…	…	…	…	…
7	7	victor	1	1,2
8	8	victor	2	3,4
9	9	victor	3	7
10	10	victor	3	7
11	11	victor	4	8
12	12	victor	4	8
13	13	victor	5	9,10,11,12
14	14	victor	5	9,10,11,12

Granularity Policy

ID	Domain	Actor ID	Subject	Condition	Policy
1	victor	2	process	process. executabl ID = AIR	Collect ALL

Fig. 4. Provenance Records for workflow in Fig 2(b)

4 Supported Queries

Having defined a comprehensive provenance model, we can use any standard query language to query the entities in the model. We discuss below the various queries supported by our provenance model:

Fundamental Queries on Entity Attributes: These queries retrieve information about the fundamental entities of the provenance model. Examples of such queries are: find all the operations belonging to a process, generate the sequence of processes/operations in a workflow. These queries can help in detecting anomalies by comparing the expected output of an operation in the recorded environment with the actual result. Users that have executed anomalous operations can be identified by finding out the actors that invoked the operations.

Queries on Invocations: These queries retrieve the set of commands involved in the manipulation of a selected data object. Users can set various filters while retrieving the provenance, such as remove commands that occurred before or after a given point of time. These queries facilitate users for reproducing a data object, detecting system changes or intrusions, finding out the system configuration during process invocation, understanding system dependencies, etc.

Queries on Lineage: The historical dependencies of a data object can be determined by traversing the provenance graph backward whereas data usage can be traced by forward traversal of the graph. A simple query is of the form: *find the ancestor data objects to data d.* More complex queries may refer to patterns within the derivation graph. The basic approach is to match specific patterns of processes consisting of operations and communications and enabling the composition of *flowpattern* objects. The flowpattern graphs can match either a fixed or varying number of nodes of their corresponding types in any workflow defined in the database. Possible queries inclue:*find the data objects that are result of a specific flowpattern*, and *find all operations in a workflow whose inputs have been processed by a specific flowpattern.*

Provenance View: Since provenance grows fast, it might be convenient (often required) to compress or summarize the provenance graph for efficient querying and navigation. For example, instead of keeping track of how different processes and people modified a document five years ago, we can replace the part of the provenance with the end result of the modification. Since our provenance model has a modularized structure, we support queries to generate any abstraction of a provenance graph. The *domain* attribute in the provenance records greatly helps in writing a quick and effective abstraction function for an intended purpose.

5 Related Work

We here review a selection of provenance-enabled systems, their underlying models and discuss their lacking in providing a generic, unified framework.

Workflow based provenance systems [1][10][11] collect provenance for data-centric workflows in a service oriented architecture. Chimera [1] defines a Virtual Data Language (VDL) to explicitly represent the workflows. In myGrid, the information model of the provenance logs contain the services invoked, their parameters, the start and end times, the data products used and derived, and ontology descriptions. Karma collects provenance at 11 activities transpired at 3 different levels, namely {*Workflow, Service, Application*} × {*-Started, -Finished, -Failed*}, *Data -Produced*, and *-Consumed*. However, all workflow based provenance models are tightly coupled to a specific system and capture provenance only at a file granularity. Cohen et al. [12] provide a generic and expressive formal model of provenance for scientific workflows.

Process based provenance systems [13] rely on individual services to record their own provenance in the form of assertions that reflect the relationships between represented services and data. In PreServ [13], a service invocation generates three types of assertions: *interaction* that records the source and sink of the service; *Actor State* with the list of input and output data of the interaction; and two *Relationship* assertions that associate the *Interaction* assertion with the produced and consumed data in the *Actor State* assertion.

PASS [4] and ES3 [3] are examples of the **OS-based provenance** approach. PASS operates at the level of shared storage system and records information about which programs are executed, their inputs, and any new files created as output. ES3 captures provenance metadata including data object identifier, domain name, input and output files. However, none of these systems provides a formal structure for provenance metadata.

From the above discussion, it is obvious that existing provenance models apply only to a particular application/domain and do not support security. Perhaps the

Table 2. Comparison between our model and existing provenance models

	Our Model	Qun Ni Model	Chimera	myGrid	Karma	PReServ	ES3	PASS
Target System	Any	Any	Workflow	Workflow	Workflow	Service	Workflow	File System
Data Granularity	Any data object	Any data object	Abstract dataset	Abstract resources	Data in a workflow	Process	File	File
Inter-operability	Yes	No	No	No	No	No	No	Yes
Security	Yes	Yes	No	No	No	No	No	No
Level of Granularity	Flexible	Rigid	Rigid	Rigid	Rigid	Rigid	Rigid	Rigid
Representation Scheme	Any	Any	VDL	XML /RDF	XML	XML	XML	Berkeley DB
Abstraction	Yes	No	Yes	No	Yes	No	No	No
Query Language	Any	Any	VDL	XML	XQuery	Custom query tool	XML	Custom query tool

provenance model by Ni et al. [5] is the most comprehensive model. However, this model documents provenance data at a granularity of operation which basically indicates functions. This fact makes it difficult to fit the model in workflow systems - composed of services with many underlying processes or in a large organization where there are multiple computing domains. Since the model does not support user specified granularity policies, the execution of a workflow will always generate a large volume of provenance records. In addition, the model does not contain a *lineage* entity to help generating separate data dependency and process dependency graphs at a fast speed.

Table 2 shows a comparison of our provenance model with other major models from various design aspects.

6 Conclusion

In this paper, we propose a comprehensive provenance model that can encapsulate the data provenance captured at different stages of a physical/computational process. We analyze the requirements for such a model and then discuss how our model meets these requirements. The model captures the characteristics of standard provenance models (e.g. OPM) and previously proposed provenance models which ensures the inter-operability of provenance across different systems.

Acknowledgements. The work reported in this paper has been partially funded by NSF under grant CNS 1111512.

References

1. Foster, I., Vöckler, J., Wilde, M., Zhao, Y.: Chimera: A virtual data system for representing, querying, and automating data derivation. In: Proc. of the Conference on Scientific and Statistical Database Management (SSDBM), pp. 37–46 (2002)

2. Janée, G., Mathena, J., Frew, J.: A data model and architecture for long-term preservation. In: Proc. of the Conference on Digital Libraries, pp. 134–144 (2008)
3. Frew, J., Metzger, D., Slaughter, P.: Automatic capture and reconstruction of computational provenance. Concurrency and Computation: Practice and Experience 20, 485–496 (2008)
4. Muniswamy-Reddy, K., Holland, D., Braun, U., Seltzer, M.: Provenance-aware storage systems. In: Proc. of the USENIX Annual Technical Conference (2006)
5. Ni, Q., Xu, S., Bertino, E., Sandhu, R., Han, W.: An Access Control Language for a General Provenance Model. In: Jonker, W., Petković, M. (eds.) SDM 2009. LNCS, vol. 5776, pp. 68–88. Springer, Heidelberg (2009)
6. Sultana, S., Bertino, E.: A comprehensive model for provenance. Poster Paper, Proc. of 4th International Provenance and Annotation Workshop, IPAW (2012)
7. Woodruff, A., Stonebraker, M.: Supporting fine-grained data lineage in a database visualization environment. In: Proc. of the International Conference on Data Engineering (ICDE), pp. 91–102 (1997)
8. Moreau, L., Clifford, B., Freire, J., et al.: The open provenance model core specification (v1.1). Future Generation Computer Systems 27(6), 743–756 (2011)
9. Huettel, S., Song, A., McCarthy, G.: Functional magnetic resonance imaging. Sinauer Associates (2004)
10. Zhao, J., Goble, C., Stevens, R., Bechhofer, S.: Semantically linking and browsing provenance logs for e-science. In: Semantics of a Networked World Semantics For Grid Databases, pp. 158–176 (2004)
11. Plale, B., Gannon, D., Reed, D., Graves, S., Droegemeier, K., Wilhelmson, B., Ramamurthy, M.: Towards Dynamically Adaptive Weather Analysis and Forecasting in LEAD. In: Sunderam, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2005. LNCS, vol. 3515, pp. 624–631. Springer, Heidelberg (2005)
12. Cohen, S., Cohen-Boulakia, S., Davidson, S.: Towards a Model of Provenance and User Views in Scientific Workflows. In: Leser, U., Naumann, F., Eckman, B. (eds.) DILS 2006. LNCS (LNBI), vol. 4075, pp. 264–279. Springer, Heidelberg (2006)
13. Groth, P., Miles, S., Moreau, L.: PReServ: Provenance recording for services. Translator (2005)

Towards Discovering Ontological Models from Big RDF Data

Carlos R. Rivero, Inma Hernández, David Ruiz, and Rafael Corchuelo

University of Sevilla, Spain
{carlosrivero,inmahernandez,druiz,corchu}@us.es

Abstract. The Web of Data, which comprises web sources that provide their data in RDF, is gaining popularity day after day. Ontological models over RDF data are shared and developed with the consensus of one or more communities. In this context, there usually exist more than one ontological model to understand RDF data, therefore, there might be a gap between the models and the data, which is not negligible in practice. In this paper, we present a technique to automatically discover ontological models from raw RDF data. It relies on a set of SPARQL 1.1 structural queries that are generic and independent from the RDF data. The output of our technique is a model that is derived from these data and includes the types and properties, subtypes, domains and ranges of properties, and minimum cardinalities of these properties. Our technique is suitable to deal with Big RDF Data since our experiments focus on millions of RDF triples, i.e., RDF data from DBpedia 3.2 and BBC. As far as we know, this is the first technique to discover such ontological models in the context of RDF data and the Web of Data.

Keywords: Ontological models, Web of Data, RDF, SPARQL 1.1.

1 Introduction

The goal of the Semantic Web is to endow the current Web with metadata, i.e., to evolve it into a Web of Data [23, 28]. Currently, there is an increasing popularity of the Web of Data, chiefly in the context of Linked Open Data, which is a successful initiative that consists of a number of principles to publish, connect, and query data in the Web [3]. Sources that belong to the Web of Data focus on several domains, such as government, life sciences, geography, media, libraries, or scholarly publications [14]. These sources offer their data using the RDF language, and they can be queried using the SPARQL query language [1].

The goal of the Web of Data is to use the Web as a large database to answer structured queries from users [23]. One of the most important research challenges is to cope with scalability, i.e., processing data at Web scale, usually referred to as Big Data [5]. Additionally, sources in the Web of Data are growing steadily, e.g., in the context of Linked Open Data, there were roughly 12 such sources in 2007 and, as of the time of writing this paper, there exist 326 sources [19]. Therefore, the problem of Big Data increases due to this large amount of sources.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 131–140, 2012.

Ontological models are used to model RDF data, and they comprise types, data properties, and object properties, each of which is identified by a URI [1]. These models are shared and developed with the consensus of one or more communities [26], which define a number of inherent constraints over the models, such as subtypes, the domains and/or ranges of a property, or the minimum and maximum cardinalities of a property.

It is important to notice that, in "traditional" information systems, developers first need to create a data model according to the user requirements, which is later populated. Contrarily, in web-of-data information systems, data can exist without an explicit model; even more, several models may exist for the same set of data. Therefore, in the context of the Web of Data, we cannot usually rely on existing ontological models to understand RDF data since there might be a gap between the models and the data, i.e., the data and the model are usually devised in isolation, without taking each other into account [11]. Furthermore, RDF data may not satisfy a particular ontological model related to these data, which is mandatory to perform a number of tasks, such as data integration [20], data exchange [25], data warehousing [12], or ontology evolution [9].

We have identified two common situations in practice in which the gap between ontological models and RDF data is not negligible, namely:

- Languages to represent ontological models provide constructs to express user-defined constraints that are local, i.e., a user or a community can add them to adapt existing models to local requirements [7]. For instance, the ontological model of DBpedia 3.7 [4], which is a community effort to make the data stored at Wikipedia accessible using the Linked Open Data principles, defines a property called *almaMater* that has type *Person* as domain, and type *EducationalInstitution* as range. It is not difficult to find out that this property has also types *City* and *Country* as ranges in the RDF data. As a conclusion, there are cases in which RDF data may not be modelled according to existing ontological models, i.e., the data may not satisfy the constraints of the models.
- Some ontological models simply define vocabularies with very few constraints. Therefore, it is expected that users of these ontological models apply them in different ways [27]. For instance, the ontological model of DBpedia 3.7 defines a property called *similar* that has neither domain nor range. In the RDF data, we observe that this property has two different behaviours: one in which type *Holiday* is the domain and range of the property, and another one in which type *Place* is the domain and range of the property. As a conclusion, different communities may generate a variety of RDF data that rely on the same ontological models with disparate constraints.

In this paper, we present a technique to automatically discover ontological models from raw RDF data. It aims to solve the gap between the models and the data. Our technique assumes that the model of a set of RDF data is not known a priori, which is a common situation in practice in the context of the Web of Data. To perform this discovery, we rely on a set of SPARQL 1.1 structural queries that are generic and independent from the RDF data, i.e., they can be applied to discover an ontological model in any set of RDF data.

The output of our technique is a model that includes the types and properties, subtypes, domains and ranges of properties, and minimum cardinalities of these properties. However, currently, we are not able to compute a number of constraints, such as subproperties, maximum cardinalities, or unions of types. Our technique is suitable to deal with Big RDF Data since our experiments focus on millions of RDF triples, i.e., RDF data from DBpedia 3.2 and BBC. To the best of our knowledge, this is the first technique to discover such ontological models in the context of RDF data and the Web of Data.

This paper is organised as follows: Section 2 describes the related work; Section 3 presents our technique to discover ontological models from RDF data that relies on a set of SPARQL 1.1 queries; Section 4 describes two experiments to discover the ontological models behind the RDF data of DBpedia 3.2 and BBC; finally, Section 5 recaps on our main conclusions.

2 Related Work

Research efforts on the automatic discovery of data models have focused on the Deep Web, in which web pages are automatically produced by filling web templates using the data of a back-end database [13]. In the context of the Web of Data, current research efforts assume that RDF data satisfy all of the constraints of the ontological models that model them; however, this situation is not so common in practice.

There are a number of proposals in the literature that aim to discover types from instances, i.e., a particular instance has a particular type. The vast majority of these proposals discover different types in web sites by clustering web pages of the same type [6, 10, 16, 21]. Mecca et al. [21] developed an algorithm for clustering search results of web sites by type that discovers the optimal number of words to classify a web page. Blanco et al. [6] devised a technique to automate the clustering of web pages by type in large web sites. The authors do not rely on the content of web pages, but only on the URLs. Hernández et al. [16] devised a technique similar in spirit to [6] technique, but using a smaller subset of web pages as the training set to automatically cluster the web pages. As a conclusion, these proposals are only able to discover types and no relationships amongst them, such as data properties, object properties, or subtypes. Giovanni et al. [10] aimed to automatically discover the untyped entities that DBpedia comprises, and they proposed two techniques based on induction and abduction.

There exist a number of proposals that are able to automatically discover the data models that are implicit in the semi-structured data that is rendered in a web page. The vast majority of these proposals focus on automating the extraction of information from these web pages [2, 8, 17], and the data models that they are able to discover comprise types and relationships amongst those types. As a conclusion, these proposals are not able to automatically infer the whole data model of the back-end database, but only a part of it.

Other proposals allow to discover complex data models that include types, properties, domains and ranges. These proposals are not fully-automated since they require the intervention of a user. Tao et al. [30] presented a proposal that automatically infers a data model by means of a form, and they deal with any kind of form, not necessarily HTML forms. In this case, the user is responsible for handcrafting these forms; unfortunately, this approach is not appealing since integration costs may be increased if the user has to intervene [22]. Furthermore, this proposal is not able to deal with subtypes.

Hernández et al. [15] devised a proposal that deals with discovering the data model behind a web site. This proposal takes a set of URL patterns that describe the types in a web site as input. Its goal is to discover properties amongst the different types that, in addition to the URL patterns of types, form a data model. The main drawback of this proposal is that it requires the intervention of the user: the final data model comprises a number of anonymous properties and the user is responsible for naming them, which may increase integration costs. In addition, this proposal is not able to discover data properties or subtypes.

Finally, Su et al. [29] developed a fully-automated proposal that discovers an ontological model that is based on the HTML forms of a web site, and the HTML results of issuing queries by means of these forms. In this case, there is no intervention of a user to discover the final ontological model, which is performed by means of a number of matchings amongst the HTML results and the HTML forms. To build the final model, the authors apply nine heuristics, such as "if a matching is unique, a new attribute is created", or "if the matching is n:1, $n + 1$ attributes are created". The main drawback of this proposal is that it does not discover subtypes or the name of the properties, i.e., the final model is more a nested-relational model than an ontological model. Note that a nested-relational model is defined by means of a tree that comprises a number of nodes, which may be nested and have a number of attributes, and it is also possible to specify referential constraints that relate these attributes [24].

3 Discovering Ontological Models

We have devised a technique that relies on a number of SPARQL 1.1 queries to discover ontological models from raw RDF data. In this section, we use a running example based on DBpedia, which has undergone several revisions. We focus on a part of DBpedia 3.2 that comprises $2,107,451$ triples, which is a dataset of Big RDF Data.

RDF data comprise triples of two kinds: type and property triples. A triple comprises three elements: the subject, the predicate, and the object, respectively. Both subjects and predicates are URIs, and objects may be URIs or literals. In the rest of this paper, we use a number of prefixes that are presented in Table 1. A type triple relates a URI with a particular type by means of a type predicate, e.g., $(dbpd{:}Clint\_Eastwood, rdf{:}type, dbpo{:}Actor)$ states that Clint Eastwood is an actor. A data property triple relates a URI with a literal using a property, e.g., $(dbpd{:}Clint\_Eastwood, dbpo{:}birthDate, \text{“}1930{-}05{-}31\text{”\^{}\^{}}xsd{:}date)$ is

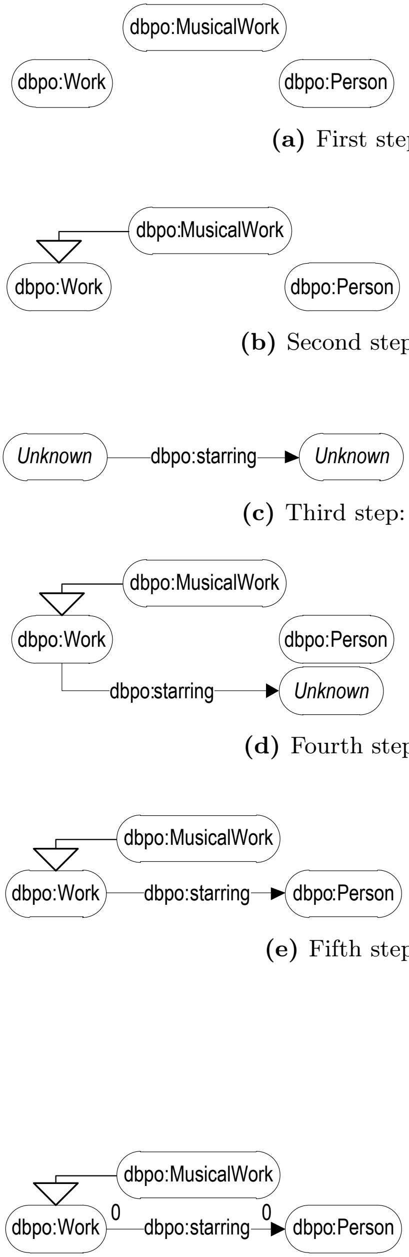
Table 1. Prefixes used throughout the paper

Prefix	URI
rdf	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`
xsd	`http://www.w3.org/2001/XMLSchema#`
dbpo	`http://dbpedia.org/ontology/`
dbpd	`http://dbpedia.org/resource/`
po	`http://purl.org/ontology/po/`
dc	`http://purl.org/dc/elements/1.1/`

a triple that states that the birth date of Clint Eastwood is May 31, 1930, which is of $xsd{:}date$ type. An object property triple relates two URIs by means of a property, e.g., $(dbpd{:}Dirty\_Harry, dbpo{:}starring, dbpd{:}Clint\_Eastwood)$ is a triple stating that film Dirty Harry is starred by Clint Eastwood.

Figure 1 presents a summary of our technique based on the ontological model of DBpedia 3.2: we first discover types and subtypes; then, we discover properties, the domains and ranges of these properties, and their minimum cardinalities. To discover this model, we issue a number of SPARQL 1.1 queries over the RDF data that are also presented in this figure, in which we enclose parameters between $ symbols. In the rest of this section, we describe each of these steps in detail:

1. In the first step, we discover types from the input RDF data, such as $dbpo{:}Person$, $dbpo{:}MusicalWork$, or $dbpo{:}Work$ (see Figure 1a). To discover them, we project the types of all instances without repetition.
2. In the second step, we discover subtypes amongst the previously discovered types. To perform this, we iterate two times over the whole set of types, so, for each pair of types t_1 and t_2, assuming that $t_1 \neq t_2$, we have that t_1 is subtype of t_2 if each instance of type t_1 is also an instance of type t_2. An example is that $dbpo{:}MusicalWork$ is subtype of $dbpo{:}Work$ (see Figure 1b). Note that we use the negation of the query in Figure 1b, i.e., t_1 is subtype of t_2 if the query returns false.
3. In the third step, we discover properties from the input RDF data, such as *dbpo*:*birthDate*, *dbpo*:*starring*, or *dbpo*:*director* (see Figure 1c). We project the predicates that relate all triples without repetition.
4. The fourth step deals with discovering domains, such as the domain of *dbpo*:*starring* is $dbpo{:}Work$ (see Figure 1d). To discover the domains of a property *prop*, we retrieve all triples that have this property as predicate, and we project the types of the subjects in these triples without repetition.
5. The fifth step is similar to the previous step, but we discover ranges instead of domains (see Figure 1e).
6. The sixth step discovers minimum cardinalities of the previously discovered domains and ranges. An example is that the minimum cardinality of *dbpo*:*starring* for domain $dbpo{:}Work$ is zero since there exists, at least,



```
SELECT DISTINCT ?t
WHERE {
    ?x  rdf:type    ?t . }
```

(a) First step: discovering types

```
ASK {
    ?s  rdf:type $t_1$ .
    FILTER NOT EXISTS {
        ?s rdf:type $t_2$ . } }
```

(b) Second step: discovering subtypes

```
SELECT DISTINCT ?p
WHERE {
    ?s  ?p  ?o . }
```

(c) Third step: discovering properties

```
SELECT DISTINCT ?d
WHERE {
    ?s   rdf:type  ?d ;
         $prop$    ?o . }
```

(d) Fourth step: discovering domains

```
SELECT DISTINCT ?r
WHERE {
    ?o   rdf:type  ?r .
    ?s  $prop$    ?o . }
```

(e) Fifth step: discovering ranges

```
ASK {
    ?s  rdf:type $type$ .
    FILTER NOT EXISTS {
        ?s  $prop$ ?o . } }
ASK {
    ?o  rdf:type $type$ .
    FILTER NOT EXISTS {
        ?s  $prop$ ?o . } }
```

(f) Sixth step: discovering cardinalities

Fig. 1. Steps of our technique to discover ontological models from RDF data

one instance of *dbpo*:*Work* that is not related by property *dbpo*:*starring* (see Figure 1f). Another example is that the minimum cardinality of property *dbpo*:*starring* for range *dbpo*:*Person* is zero since there exists, at least, one instance of *dbpo* : *Person* that is not the subject of an instance of property *dbpo* : *starring*. To perform this, we ask if there is any domain or range instance of a given type *type* related by a particular property *prop*. If this is true, the minimum cardinality is zero. Otherwise, we count the minimum number of instances of type *type* related to property *prop*.

Our technique is not able to discover a number of constraints, but some of them may be addressed, e.g., maximum cardinalities and subproperties. Regarding maximum cardinalities, our technique is able to compute a bound of the cardinality, but not the exact cardinality. For instance, we have computed that the maximum cardinality of property *dbpo*:*starring* for domain *dbpo*:*Work* is 74, however, this number is not the exact cardinality since it probably allows unbounded instances. Regarding subproperties, we may use a technique similar to the second step to discover subtypes.

4 Experiment Results

We implemented our technique using Java 1.6 and OWLIM Lite 4.2, which comprises an RDF store and a SPARQL query engine. In this experiment, we computed the times taken by our technique to discover the ontological models behind the RDF data of a part of DBpedia 3.2 and BBC. The BBC [18] decided to adhere to the Linked Open Data principles in 2009. They provide ontological models that adhere to these principles to publicise the music and programmes they broadcast in both radio and television.

To compute the times taken by our technique, we ran the experiment on a virtual computer that was equipped with a four-threaded Intel Xeon 3.00 GHz CPU and 16 GB RAM, running on Windows Server 2008 (64-bits), JRE 1.6.0. Furthermore, we repeated the experiment 25 times and computed the maximum values. Table 2 shows our results when applying our technique to DBpedia 3.2 and BBC. The first column of the table stands for the different steps of our technique; the second column deals with the total number of constraints that we have discovered; finally, the third column shows the time in minutes taken by our technique to compute each step.

The total time that our technique took was 31.15 minutes for DBpedia 3.2, which comprises a total number of 2, 107, 451 triples, and 2.48 minutes for BBC, which comprises a total number of 7, 274, 597 triples. At a first glance, it might be surprising that the time taken for BBC is less than the time taken for DBpedia, since BBC comprises more triples than DBpedia. This is due to the fact that the time of our technique depends on the structural complexity of the discovered ontological model, and it does not depend on the data. Therefore, we may conclude that the structural complexity of the ontological model of DBpedia 3.2 is greater than the structural complexity of the BBC model.

Table 2. Summary of results of discovering ontological models behind RDF data

(a) DBpedia 3.2

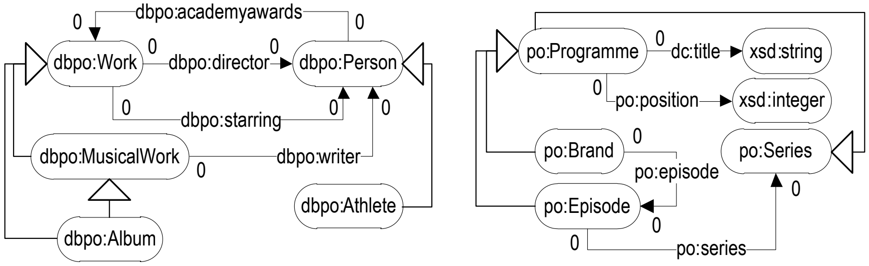
Step	Constr.	Time (min)
Types	91	0.03
Subtypes	328	0.12
Properties	398	0.02
Domains	1, 148	16.15
Ranges	1, 148	12.35
Cardinalities	4, 592	2.48
Total	7, 705	31.15

(b) BBC

Step	Constr.	Time (min)
Types	15	0.03
Subtypes	6	0.17
Properties	28	0.05
Domains	39	0.99
Ranges	39	1.09
Cardinalities	156	0.15
Total	283	2.48

Figure 2a shows a part of the ontological model that results from applying our technique to the RDF data of DBpedia 3.2. In this case, the model comprises five types, namely: *dbpo*:*Person*, *dbpo*:*Work*, *dbpo*:*Athlete*, *dbpo*:*MusicalWork*, and *dbpo*:*Album*. In addition to these types, the model comprises four subtype relationships, and four properties with their domains and ranges, namely: *dbpo*:*starring*, *dbpo*:*director*, *dbpo*:*writer*, and *dbpo*:*academyawards*. Finally, the minimum cardinalities for all properties are zero.

Figure 2b shows a part of the model that results from the RDF data of BBC, which comprises four types, namely: *po*:*Programme*, *po*:*Brand*, *po*:*Episode*, and *po*:*Series*. It also comprises three subtype relationships, and four properties with their domains and ranges, namely: *dc*:*title*, *po*:*position*, *po*:*episode*, and *po*:*series*. Note that the minimum cardinalities for all properties are also zero.



(a) DBpedia 3.2 **(b)** BBC

Fig. 2. A part of the ontological models that result from our experiments

5 Conclusions

In the context of the Web of Data, there exists a gap between existing ontological models and RDF data due to the following reasons: 1) RDF data may not satisfy the constraints of the existing ontological models; 2) different communities may generate a variety of RDF data that rely on the same ontological models with disparate constraints. This gap is not negligible and may hinder the practical application of RDF data and ontological models in other tasks, such as data integration, data exchange, data warehousing, or ontology evolution. To solve this gap, we present a technique to discover ontological models from raw RDF data that relies on a set of SPARQL 1.1 structural queries. The output of our technique is a model that includes types and properties, subtypes, domains and ranges of properties, and minimum cardinalities of these properties.

Acknowledgements. Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

References

[1] Antoniou, G., van Harmelen, F.: A Semantic Web Primer. The MIT Press (2008)

[2] Arasu, A., Garcia-Molina, H.: Extracting structured data from web pages. In: SIGMOD Conference, pp. 337–348 (2003)

[3] Bizer, C., Heath, T., Berners-Lee, T.: Linked Data: The story so far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)

[4] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. J. Web Sem. 77(3), 154–165 (2009)

[5] Bizer, C., Boncz, P., Brodie, M.L., Erling, O.: The meaningful use of Big Data: Four perspectives - four challenges. SIGMOD Record 40(4), 56–60 (2011)

[6] Blanco, L., Dalvi, N.N., Machanavajjhala, A.: Highly efficient algorithms for structural clustering of large websites. In: WWW, pp. 437–446 (2011)

[7] Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: Contextualizing ontologies. J. Web Sem. 1(4), 325–343 (2004)

[8] Crescenzi, V., Mecca, G.: Automatic information extraction from large websites. J. ACM 51(5), 731–779 (2004)

[9] Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. Knowledge Eng. Review 23(2), 117–152 (2008)

[10] Giovanni, A., Gangemi, A., Presutti, V., Ciancarini, P.: Type inference through the analysis of wikipedia links. In: LDOW (2012)

[11] Glimm, B., Hogan, A., Krötzsch, M., Polleres, A.: OWL: Yet to arrive on the Web of Data? In: LDOW (2012)

[12] Glorio, O., Mazón, J.-N., Garrigós, I., Trujillo, J.: A personalization process for spatial data warehouse development. Decision Support Systems 52(4), 884–898 (2012)

[13] He, B., Patel, M., Zhang, Z., Chang, K.C.-C.: Accessing the Deep Web. Commun. ACM 50(5), 94–101 (2007)

[14] Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool (2011)

[15] Hernández, I., Rivero, C.R., Ruiz, D., Corchuelo, R.: Towards Discovering Conceptual Models behind Web Sites. In: Atzeni, P., Cheung, D., Sudha, R. (eds.) ER 2012. LNCS, vol. 7532, pp. 166–175. Springer, Heidelberg (2012)

[16] Hernández, I., Rivero, C.R., Ruiz, D., Corchuelo, R.: A statistical approach to URL-based web page clustering. In: WWW, pp. 525–526 (2012)

[17] Kayed, M., Chang, C.-H.: FiVaTech: Page-level web data extraction from template pages. IEEE Trans. Knowl. Data Eng. 22(2), 249–263 (2010)

[18] Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., Lee, R.: Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 723–737. Springer, Heidelberg (2009)

[19] LOD Cloud. Linked Open Data cloud (April 2012), http://thedatahub.org/group/lodcloud

[20] Makris, K., Gioldasis, N., Bikakis, N., Christodoulakis, S.: SPARQL-RW: Transparent query access over mapped RDF data sources. In: EDBT (2012)

[21] Mecca, G., Raunich, S., Pappalardo, A.: A new algorithm for clustering search results. Data Knowl. Eng. 62(3), 504–522 (2007)

[22] Petropoulos, M., Deutsch, A., Papakonstantinou, Y., Katsis, Y.: Exporting and interactively querying web service-accessed sources: The CLIDE system. ACM Trans. Database Syst. 32(4), 22 (2007)

[23] Polleres, A., Huynh, D.: Special issue: The Web of Data. J. Web Sem. 7(3), 135 (2009)

[24] Popa, L., Velegrakis, Y., Miller, R.J., Hernández, M.A., Fagin, R.: Translating web data. In: VLDB, pp. 598–609 (2002)

[25] Rivero, C.R., Hernández, I., Ruiz, D., Corchuelo, R.: On benchmarking data translation systems for semantic-web ontologies. In: CIKM, pp. 1613–1618 (2011)

[26] Rivero, C.R., Hernández, I., Ruiz, D., Corchuelo, R.: Generating SPARQL Executable Mappings to Integrate Ontologies. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 118–131. Springer, Heidelberg (2011b)

[27] Rivero, C.R., Schultz, A., Bizer, C., Ruiz, D.: Benchmarking the performance of Linked Data translation systems. In: LDOW (2012)

[28] Shadbolt, N., Berners-Lee, T., Hall, W.: The Semantic Web revisited. IEEE Intelligent Systems 21(3), 96–101 (2006)

[29] Su, W., Wang, J., Lochovsky, F.H.: ODE: Ontology-assisted data extraction. ACM Trans. Database Syst. 34(2), 12 (2009)

[30] Tao, C., Embley, D.W., Liddle, S.W.: FOCIH: Form-Based Ontology Creation and Information Harvesting. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 346–359. Springer, Heidelberg (2009)

Towards Scalable Information Modeling of Requirements Architectures

Krzysztof Wnuk[1], Markus Borg[1], and Saïd Assar[1, 2]

[1] Department of Computer Science, Lund University, Lund Sweden
[2] Telecom Ecole de Management, France
{Krzysztof.Wnuk,Markus.Borg}@cs.lth.se,
said.assar@it-sudparis.eu

Abstract. The amount of data in large-scale software engineering contexts continues to grow and challenges efficiency of software engineering efforts. At the same time, information related to requirements plays a vital role in the success of software products and projects. To face the current challenges in software engineering information management, software companies need to reconsider the current models of information. In this paper, we present a modeling framework for requirements artifacts dedicated to a large-scale market-driven requirements engineering context. The underlying meta-model is grounded in a clear industrial need for improved flexible models for storing requirements engineering information. The presented framework is created in collaboration with industry and initially evaluated by industry practitioners from three large companies. Participants of the evaluation positively evaluated the presented modeling framework as well as pointed out directions for further research and improvements.

Keywords: Large-scale requirements engineering, requirements architectures, empirical study, requirements modeling.

1 Introduction

Requirements engineering is an important part of the software development lifecycle as it helps to identify what should be implemented in software products to make them successful. As a knowledge intense part of the software development process, requirements engineering contributes to the generation of large amounts of information that need to be managed.

The size and complexity of software engineering artifacts continues to grow as a result of increasing complexity of software intensive systems. As a result, software development companies that operate globally often have to face the challenges of storing over 10 000 requirements in the requirements database [2,4]. The amount of information to manage increases even more if we consider additional software development information such as product strategies, design documents, test case descriptions and defect reports.

In a recent study, we introduced a classification of requirements engineering contexts based on the number of requirements and the number of interdependencies

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 141–150, 2012.

between requirements as a proxy for complexity [2]. We defined Very-Large Scale Requirements Engineering (VLSRE) as a context where the number of requirements and interdependencies exceeds 10 000 and manually managing a complete set of interdependencies among small bundles of requirements is unfeasible in practice. While empirically exploring challenges in VLSRE, we discovered that one of the challenges in VLSRE is to define and properly manage structures of requirements information, also called requirements architectures [10]. Defining a model for requirements related information could help in this and other related challenges of VLSRE. The challenge lies not only in dealing with the heterogeneity of artifacts structure that need to be managed all along the software project, but also in dealing with the frequent evolution of these structures during the lifetime of the software project.

In this paper we present a general modeling framework for requirements information in VLSRE projects created in close collaboration with industry. The underlying meta-model can describe not only requirements, but also any other pieces of relevant software development information, as suggested by our industry partners. The novelty of the approach lies in its capacity to explicitly involve external sources of information and in handling the temporal aspect related to the evolution of artifacts' structures. We conducted an initial validation of our approach with 5 practitioners from 3 companies to collect feedback, opinions and improvement proposals regarding the framework. All five respondents positively evaluated the general usefulness of the approach and provided insights and suggestions for further development and improvement of the modeling framework.

This paper is structured as follows: section 2 presents background, related work, outlines an example industrial context based of one of our industrial partners and explains the need for creating the modeling framework. Section 3 presents the research design of the study. Section 4 presents the modeling framework while section 5 presents the results of the initial evaluation of the model with industry practitioners. Section 6 discusses the limitations of the model, outlines future work and concludes the paper.

2 Large-Scale Requirements Engineering and Information Landscape on an Empirical Example

Most work in an enterprise is accompanied by some form of knowledge representation in documents [11]. Documentation is fundamental in requirements engineering, as the lack of complete specifications is a major cause of project failures [12]. However, storing too much documentation risks burdening employers by an ever-increasing amount of information. *Information overload* occurs when an individual's information processing capabilities are exceeded by the information processing requirements, i.e. the individual does not have enough time or capability to process all presented information [14]. Several studies have found that the support for decision-making is positively correlated to the amount of presented information up to a certain point, and then it declines [13, 15, 16].

In software engineering projects, large amounts of formal and informal information is continuously produced and modified [5, 6]. Thus, an important characteristic of artifacts' information in software engineering projects is its findability, defined as "the degree to which a system or environment supports navigation and retrieval" [7]. Information seeking is an increasingly costly activity among knowledge workers in general [8]. Software engineering projects are no exceptions, as identified by previous case studies in this context [5, 9].

We present an example of a VLSRE context based on a longitudinal study we have been conducting at a large company since fall 2007. Focusing on feature tracking, we observed the structure of information related to product features and the associated detailed requirements, and the evolution of this structure over time and over projects. Together with observing the evolution of the information structure, we have in fall 2007 conducted 7 in-depth interviews to understand the role of information structures and their impact on the VLSRE context. Partial results from this study were published in [3, 10]. During these 7 interviews, we have conceptualized the following picture of the information landscape while managing requirements and features in a VLSRE context, see Fig.1.

In an example of a VLSRE context, we have key customers submitting their requirements specifications to the requirements repository, and suppliers receiving specifications based on interpretations of these key customers' wishes and market trends. Special teams of experts (STEs) together with product planning, assisted by requirements analysts and business analysts, create natural language descriptions of candidate future software features. The features are later refined by STEs to a set of more detailed system requirements and merged into the current requirements architecture. As it is indicated in Fig. 1, every new specification deliverable contains partial requirements structures that should fit within requirements architecture and be merged with the requirements repository.

The current number of features in the repository exceeds 8000 and the number of attributes associated with the features exceeds 50. Thus, the amount of information to manage is substantial. Moreover, the efficiency of requirements engineering and software development efforts depend on the accuracy, understandability and cohesion, robustness, extensibility and flexibility of the information structure [3].

3 Research Design

To evaluate a modeling method or technique, a large set of approaches are possible such as feature comparison, conceptual investigation or empirical evaluation [17]. This study adopts an empirical perspective and has been conducted in an action research mode. In action research studies, researchers make an attempt to solve a real-word problem. At the same time, researchers investigate the experiences and problems encountered while trying to solve the problem [18]. In our case, a need for developing a model for requirements information was stated by our industry partners during the interviews in 2007. Following that authentic need, we have conducted several unstructured brainstorming sessions and discussion meetings with our industry partners where we further discussed the need for the model and the high-level content of it. Moreover, we have studied the current information models used at the case company and identified their strong and weak points that were also discussed during the brainstorming sessions. Based on the result of these empirical investigations, we propose a framework for requirements information modeling presented in section 4. This framework exploits a traceability meta-models developed previously [20].

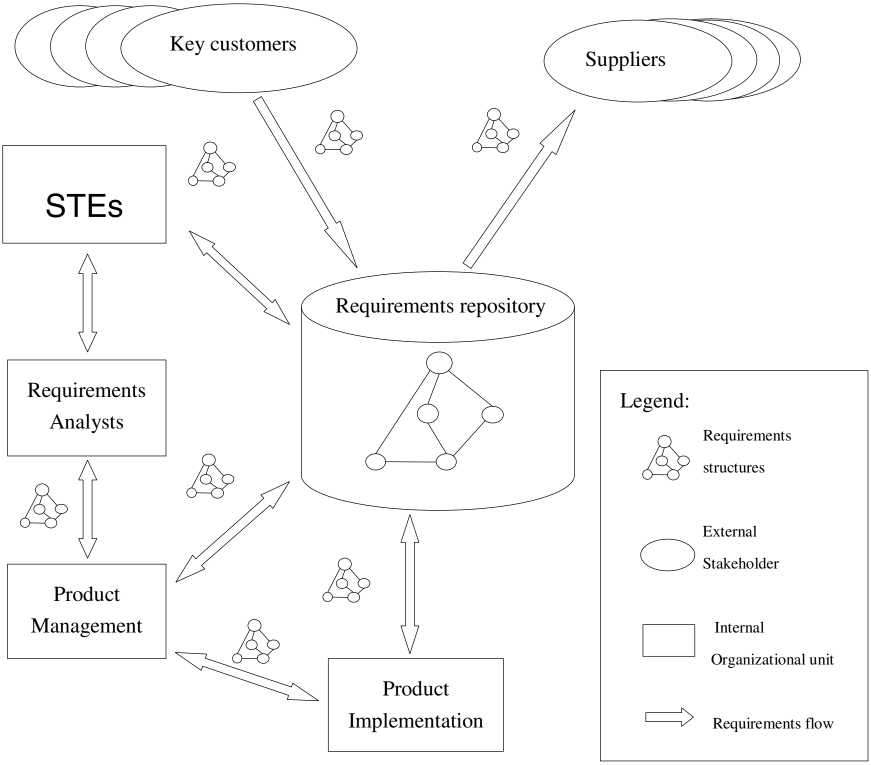


Fig. 1. An example of requirements engineering information flow in a VLSRE context

We conducted 5 interviews at 3 companies to perform the initial validation of the model. The interviews were semi-structured which means that there was a possibility to discuss aspects not covered by the interview instrument [1]. Each interview took up to 60 minutes and was attended by one researcher; who moderated the discussion and took extensive notes; and one interviewee. At the beginning of each interview, the research goals and objectives were outlined to the interviewees. Next, we discussed the information model. Further, specific questions regarding the general usefulness of the modeling framework followed by specific questions regarding the elements of the underlying meta-model were asked. Finally, we collected the interviewees' opinions regarding the limitations of the model and suggestions for improvements of the modeling framework.

4 The iMORE Framework

The core of the iMORE (*information Modeling in Requirements Engineering*) framework is the distinction between the external information structures and internal information structures, outlined in Fig. 2 by a dashed line. The importance of including external information structures was stressed several times by our industrial practitioners during the development of the modeling framework. This need for external information structures is caused by several sources of requirements and other information types that directly interact with the company, including competitors, suppliers, open source components and other partners. For all abstraction levels of the model, there is a need to be able to access external information while managing companies' internal information. For example, while looking at the source code, developers could check similar or associated open source solutions.

The structures of information are divided into three main blocks: the upstream, the requirements and the downstream blocks. In the 'upstream block' all 'high-level' information is stored, including the goals, strategies and business needs. In the 'requirements block' all requirements associated information is stored, including functional requirements, quality requirements, constraints, legal requirements and regulations. In the 'downstream' block the information related to the source code, is placed, including bug reports, code documentation, and the source code itself.

The last main element in the iMORE framework is handling temporal aspect of the information structure, depicted in the vertical arrow in Fig 2. The temporal aspects include capturing the evolution of the data models in terms of the evolution of the artifacts and their associated structures. To deal with this issue, the underlying meta-model defines 'Evolution' type of links between two artifacts. Using this category of links, users can handle the evolution over time of artifacts and their structure.

The information structure in each of the blocks is defined according to a simple traceability meta-model derived from related works [19] and previous research [20]. In this meta-model (Fig. 3), the structure of an element to be stored in the repository and to be traced in the software project is constructed using two generic concepts: artifact and attribute. An attribute can be an atomic element of information (e.g. owner, release date, version number, URL) or any complex structure (e.g. list of modification dates). The set of attributes is not only limited to a particular block of information but may also cover several blocks or even the entire information structure creating a set of 'global' attributes.

According to the user needs, any artifact in the repository can be linked to other artifacts. Five categories of links are predefined in the iMORE meta-model; they are briefly explained using the following examples:

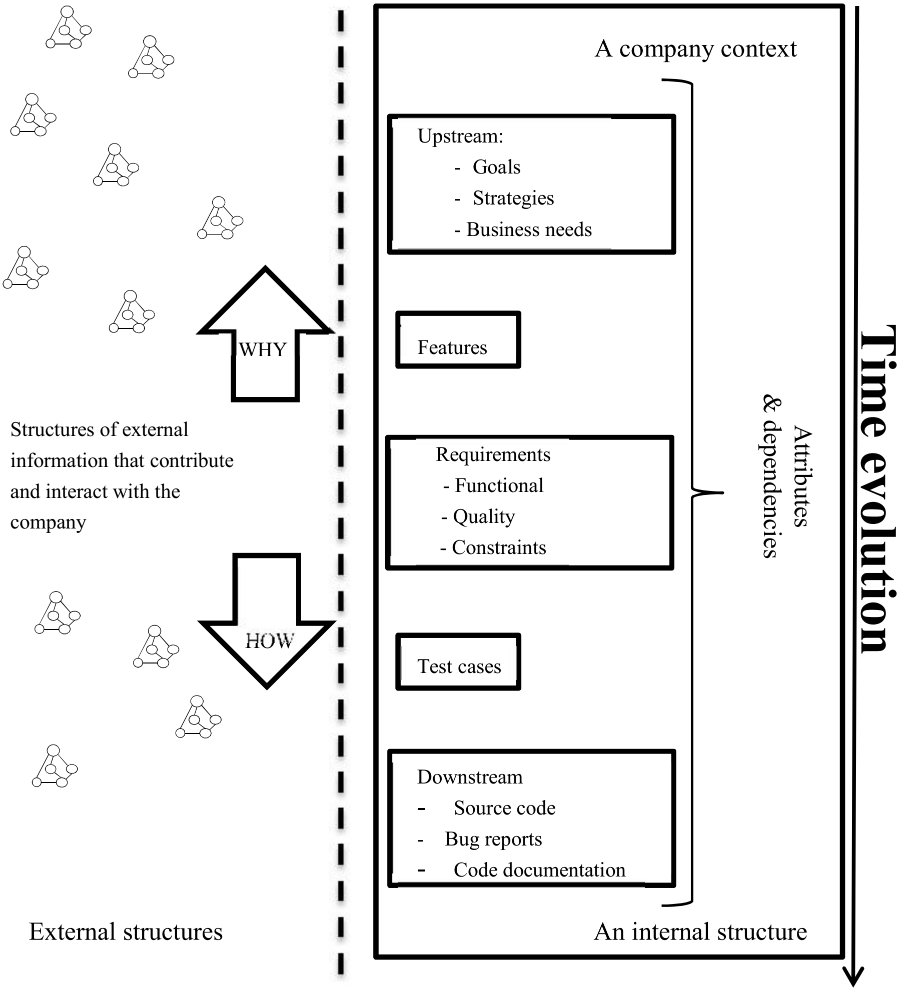


Fig. 2. The iMORE modeling framework

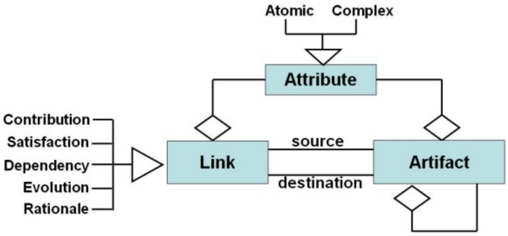


Fig. 3. The iMORE meta-model

- A requirement document A **contributes** to the specification of a design feature B
- A design feature A **satisfies** an external law based constraint B
- A design feature A **depends** on another design feature B
- A design specification A is the result of the **evolution** of a design specification B
- An external law based constraint A is the **rationale** for a requirement document B.

These links are exploited in order to find linked elements and to navigate in the repository. If systematically provided by the users, such links can contribute to a full traceability system. Such pre-traceability is often difficult to implement [21], and recent works in requirements traceability advocate combining it with post-traceability based on information retrieval techniques [22]. However, full pre- and post-traceability is not the main goal of this proposal.

5 Discussion of the iMORE Modeling Framework with Practitioners

We present the discussion of the iMORE approach based on five interviews conducted at three companies. During the interviews, we discussed not only the iMORE approach but also the relationships between the suggested meta-model and the challenges our practitioners face in their daily work. The results are outlined according to the interview instrument that can be accessed online [1].

The need for requirements information modeling. All five respondents confirmed the need for modeling requirements information in a more findable and understandable way. One respondent stressed that the need depends on the size of the company indicating that it is much more important for larger projects and companies to have effective requirements architectures in place. Another respondent indicated that the current rather high-level model has limited application and is more suited for high-level roles. Further, the same respondent stressed that the model may help to perform cross-analysis between the projects. Finally, one respondent stressed that the main goal for developing the model is to get better understanding of the knowledge of the market needs and other ‘upstream’ information.

The distinction between the internal and external information in the iMORE approach. Five respondents agreed to the distinction and stressed that external information currently dominated their daily work. Among the types of external information that our respondents need to browse are: standards and regulations, open source code and documentation, marketing resources available on the Internet etc. One respondent indicated that integrating regulations and laws to the model will be counterproductive and it will make the model hard to manage as regulations and laws can change frequently. Two other respondents mentioned that they access open source project information very often since their software product is mostly based on that solution. Those respondents also indicated that full integration of external open source project information is practically unfeasible as these projects change frequently. Further, one respondent indicated that external information is very important when "developing global services for large customers" which confirms our pre-understanding of the importance of external sources of information for projects in VLSRE. Finally, one respondent valued market and business related external information as the most valuable among the external sources. All respondent confirmed that in a large-scale MDRE context improved integration with external sources of information is important and desired.

Representation of attributes and dependencies in the iMORE approach. Two respondents agreed to the idea of separating attributes and dependencies from the requirements information. On the other hand, one respondent disagreed with this idea. Two respondents suggested that dependencies between requirements information elements are also a type of an attribute. Also, one respondent suggested that a "period of validity" attribute should be added. This attribute will improve managing the temporal aspect of the model by giving the engineers triggers and reminders about information becoming outdated that requires their attention. Another respondent indicated that the only important dependencies are one-way relations from visions to requirements and to code. Finally, one respondent suggested reusing patterns from data modeling to investigate which attributes are shared and which are unique to an instance.

Managing the temporal aspect of the information structure. Surprisingly, one respondent indicated that managing the temporal aspect of the information structure isn't so important. Another respondent suggested managing the temporal aspects of the information structure by creating an attribute for every entity called "period of validity". After the period of validity expires the information would need to be updated or deleted. Two respondents suggested implementing a similar system for managing changes based on triggers generated by changes to selected important attributes and entities in the information structure. Finally, one respondent suggested a method based on combining baselines and trigger-based updates. When it is important, a snapshot of external information should be taken and kept until it is no longer relevant. There should be a mechanism of finding the differences between the snapshot and the state of the information structure at the time when the snapshot became out of date. Changes to selected important entities of information should trigger actions, for example notification of substantial changes to the code base as oppose to bug fixes.

6 Conclusions and Further Work

Concise and quick access to information is critical in large organizations. Information overload impedes knowledge workers both in decision making and information seeking. In large-scale software development, challenging amounts of information are produced and modified throughout the entire development lifecycle. Successful management of requirements is one central activity that demands a robust information model. Increased dependence on external sources of information further stresses the situation. Thus, providing an efficient modeling framework could limit the consequences of information overload in software development projects.

In this paper, we present a modeling framework for requirements-related information for very-large market-driven requirements engineering contexts. The main novelty of our model lies in involving external sources of information and stressing the temporal aspect of the model. We evaluated our proposed with five industry practitioners from three companies. All respondents agreed with the main ideas behind the model. Moreover, they acknowledged that keeping and updating information structures in large scale software development projects is difficult. This finding is in line with previous research in knowledge intensive organizations in general [8] and the software development context in particular [5,9]. Also, our respondents confirmed that including external sources of information and the temporal aspect are strengths of our modeling framework.

Regarding the place of the attributes in the model our respondents gave inconsistent answers. However, attributes were considered as a way of handling the changes of the information structure over time. When queried about ways of managing the time evolution of the model, our respondents suggested creating an attribute for every entity called 'period of validity'. In order to handle such needs, our approach is based on meta-modeling so that information structures can be defined, modified and managed all along the software project timeline. Our approach is in line with similar works that recognize the important role of abstraction meta-levels in dealing with information interoperability and traceability in software projects [23,24]. From an implementation perspective, it was suggested managing changes in the repository using triggers or by combining baselines and triggers together.

Future works is projected in two directions. First, the iMORE framework presented here can be seen as a set of high level requirement for information architecture and management in VLSRE context. As such, it can form the basis of an evaluation framework for studying and assessing existing information management tools for software engineering (e.g. Rational RequisitePro). A second direction is to further explore stakeholders' requirements concerning information management and integration in very large software projects. This would include enhancing and validating the information meta-model that is sketched in this paper.

References

1. Wnuk, K.: The interview instrument can be accessed at (2012), `http://serg.cs.lth.se/fileadmin/serg/II.pdf`
2. Regnell, B., Svensson, R.B., Wnuk, K.: Can We Beat the Complexity of Very Large-Scale Requirements Engineering? In: Paech, B., Rolland, C. (eds.) REFSQ 2008. LNCS, vol. 5025, pp. 123–128. Springer, Heidelberg (2008)
3. Wnuk, K., Regnell, B., Berenbach, B.: Scaling Up Requirements Engineering – Exploring the Challenges of Increasing Size and Complexity in Market-Driven Software Development. In: Berry, D., Franch, X. (eds.) REFSQ 2011. LNCS, vol. 6606, pp. 54–59. Springer, Heidelberg (2011)
4. Berenbach, B., Paulish, D.J., Kazmeier, J., Rudorfer, A.: Software & Systems Requirements Engineering: In Practice. McGraw-Hill, New York (2009)
5. Olsson, T.: Software Information Management in Requirements and Test Documentation. Licentiate Thesis. Lund University, Sweden (2002)
6. Cleland-Huang, J., Chang, C.K., Christensen, M.: Event-based traceability for managing evolutionary change. Trans. Soft. Eng. 29, 796–810 (2003)

7. Morville. P.: Ambient Findability: What We Find Changes Who We Become. O'Reilly Media (2005)
8. Karr-Wisniewski, P., Lu, Y.: When more is too much: Operationalizing technology overload and exploring its impact on knowledge worker productivity. Computers in Human Behavior 26, 1061–1072 (2010)
9. Sabaliauskaite, G., Loconsole, A., Engström, E., Unterkalmsteiner, M., Regnell, B., Runeson, P., Gorschek, T., Feldt, R.: Challenges in Aligning Requirements Engineering and Verification in a Large-Scale Industrial Context. In: Wieringa, R., Persson, A. (eds.) REFSQ 2010. LNCS, vol. 6182, pp. 128–142. Springer, Heidelberg (2010)
10. Wnuk, K., Regnell, B., Schrewelius, C.: Architecting and Coordinating Thousands of Requirements – An Industrial Case Study. In: Glinz, M., Heymans, P. (eds.) REFSQ 2009. LNCS, vol. 5512, pp. 118–123. Springer, Heidelberg (2009)
11. Zantout, H.: Document management systems from current capabilities towards intelligent information retrieval: an overview. Int. J. Inf. Management. 19, 471–484 (1999)
12. Gorschek, T., Svahnberg, M., Tejle, K.: Introduction and Application of a Lightweight Requirements Engineering Process Evaluation Method. In: Proc. of the 9th Int. Workshop on Requirements Eng.: Foundation for Software Quality (REFSQ 2003), pp. 101–112 (2003)
13. Swain, M.R., Haka, S.F.: Effects of information load on capital budgeting decisions. Behavioral Research in Accounting 12, 171–199 (2000)
14. Eppler, M., Mengis, J.: The Concept of Information Overload - A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. The Information Society 20, 325–344 (2004)
15. Chewning Jr., E.C., Harrell, A.M.: The effect of information load on decision makers' cue utilization levels and decision quality in a financial distress decision task. Accounting, Organizations and Society 15, 527–542 (1990)
16. Cook, G.J.: An empirical investigation of information search strategies with implications for decision support system design. Decision Sciences 24, 683–699 (1993)
17. Siau, K., Rossi, M.: Evaluation techniques for systems analysis and design modeling methods – a review and comparative analysis. Inf. Systems Journal 21(3), 249–268 (2011)
18. Easterbrook, S., Singer, J., Storey, M.-A., Damian, D.: Selecting Empirical Methods for Software Engineering Research. In: Shull, F., et al. (eds.) Guide to Advanced Empirical Software Engineering, pp. 285–311. Springer, Heidelberg (2008)
19. Ramesh, B., Jarke, M.: Toward reference models for requirements traceability. IEEE Transactions on Software Engineering 27(1), 58–93 (2001)
20. El Ghazi, H., Assar, S.: A multi view based traceability management method. In: 2nd Int. Conf. on Research Challenges in Inf. Science, pp. 393–400. IEEE Computer Society (2008)
21. Cleland-Huang, J., Settimi, R., Romanova, E., Berenbach, B., Clark, S.: Best Practices for Automated Traceability. Computer 40(6), 27–35 (2007)
22. Borg, M., Pfahl, D.: Do better IR tools improve the accuracy of engineers' traceability recovery? In: Int. Workshop on Machine Learning Technologies in Soft. Eng., pp. 27–34 (2011)
23. Terzi, S., Cassina, J., Panetto, H.: Development of a Metamodel to Foster Interoperability along the Product Lifecycle Traceability. In: Konstantas, D., Bourrières, J.-P., Léonard, M., Boudjlida, N., et al. (eds.) Interoperability of Enterprise Software and Applications, pp. 1–11. Springer, London (2006)
24. Cavalcanti, Y.C., do Carmo Machado, I., da Mota, P.A., Neto, S., Lobato, L.L., de Almeida, E.S., de Lemos Meira, S.R.: Towards metamodel support for variability and traceability in software product lines. In: Proc. of the 5th VaMoS Workshop. ACM, NY (2011)

Preface to MORE-BI 2012

Ivan J. Jureta[1], Stéphane Faulkner[1], and Esteban Zimányi[2]

[1] University of Namur, Belgium
[2] Université Libre de Bruxelles, Belgium

The series of International Workshops on Modeling and Reasoning for Business Intelligence (MORE-BI) aims at advancing the engineering of Business Intelligence (BI) systems. The second edition of the workshop was collocated with the 31st International Conference on Conceptual Modeling (ER 2012), held in Florence, Italy, in October 2012.

BI systems gather, store, and process data to turn it into information relevant for decision-making. Successful engineering, use, and evolution of BI systems require a deep understanding of the requirements of decision-making processes in organizations, of the kinds of information used and produced in these processes, of the ways in which information can be obtained through acquisition and reasoning on data, of the transformations and analyses of that information, of how the necessary data can be acquired, stored, and cleaned, of how its quality can be improved, and of how heterogeneous data can be used together.

The second edition of MORE-BI focused on three topics: the modeling of point-based sequential data towards its analysis in an OLAP-like manner, the documentation via BI Analysis Graphs of how human analysts draw conclusions from data delivered via a BI system, and the use of foundational ontologies for ontology alignment. We hope that the workshop has stimulated discussions and contributed to the research on the concepts and relations relevant for the various steps in the engineering of BI systems.

Prof. Michael Schrefl, Department of Business Informatics - Data & Knowledge Engineering, at Johannes Kepler University of Linz, Austria held the keynote, entitled "Modelling and Reasoning Issues in SemCockpit". SemCockpit is an ontology-driven, interactive BI tool for comparative data analysis. Prof. Schrefl gave a general overview of SemCockpit, and focused on modelling and reasoning issues not addressed in other specific talks by the SemCockpit team at ER 2012 conference and MORE-BI workshop.

We thank all authors who have submitted their research to MORE-BI 2012. We are grateful to our colleagues in the steering committee for helping us define the topics and scope of the workshop, our colleagues in the program committee for the time invested in carefully reviewing the submissions under a very tight schedule, the participants who have helped make this a relevant event, and the local organizers and workshop chairs of ER 2012.

We hope that you find the workshop program and presentations of interest to research and practice of business intelligence, and that the workshop has allowed you to meet colleagues and practitioners focusing on modeling and reasoning for business intelligence. We look forward to receive your submissions and meet you at the next edition of the workshop.

OLAP-Like Analysis of Time Point-Based Sequential Data[*]

Bartosz Bębel, Mikołaj Morzy, Tadeusz Morzy,
Zbyszko Królikowski, and Robert Wrembel

Poznań University of Technology, Institute of Computing Science, Poznań, Poland
{Bartosz.Bebel,Mikolaj.Morzy,Tadeusz.Morzy,
Zbyszko.Krolikowski,Robert.Wrembel}@put.poznan.pl

Abstract. Nowadays business intelligence technologies allow to analyze mainly set oriented data, without considering order dependencies between data. Few approaches to analyzing data of sequential order have been proposed so far. Nonetheless, for storing and manipulating sequential data the approaches use either the relational data model or its extensions. We argue that in order to be able to fully support the analysis of sequential data, a dedicated new data model is needed. In this paper, we propose a formal model for time point-based sequential data with operations that allow to construct sequences of events, organize them in an OLAP-like manner, and analyze them. To the best of our knowledge, this is the first formal model and query language for this class of data.

1 Introduction

Multiple applications generate huge sets of ordered data. Some typical examples include: workflow systems, user navigation through web pages, diseases curing, RFID-based goods transportation systems (e.g., [10]), public transportation infrastructures (e.g., [2,1,15]), and remote media consumption measurement installations (e.g., [12]). Some of the data have the character of events that last an instant - a chronon, whereas some of them last for a given time period - an interval. In this regard, sequential data can be categorized either as *time point-based* or *interval-based* [16], but for all of them the order in which they were generated is important.

Since over 20 years, data analysis has been performed by means of business intelligence (BI) technologies [5] that include a data warehouse (DW) system architecture and the set of tools for advanced data analysis – the on-line analytical processing (OLAP) applications (e.g., sales trend analysis, trend prediction, data mining, social network analysis). Traditional DW system architectures have been developed in order to efficiently analyze data that originally are coming from heterogeneous and distributed data sources, maintained within an enterprise. OLAP applications, although very advanced ones, allow to analyze mainly set oriented data, but they are not capable of exploiting existing order among data. For this reason, a natural extension to traditional OLAP tools

[*] This work was supported from the Polish National Science Center (NCN), grant No. 2011/01/B/ST6/05169.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 153–161, 2012.

has been proposed in the research literature as the set of techniques and algorithms allowing to analyze data that have sequential nature, e.g., [7,8,10,11,14,15]. For storing and manipulating sequential data, the approaches use either the relational data model or its extension. We argue that in order to be able to fully support the analysis of sequential data, a dedicated new data model is needed.

Paper contribution. In this paper, we extend the draft of a formal model for time point-based sequential data [3] with the definitions of a fact, measure, dimension, and a dimension hierarchy. Thus, the model allows to analyze sequential data in an OLAP-like manner. To the best of our knowledge, this is the first comprehensive model and query language for this class of data.

2 Leading Example

As an illustration of sequential data and their analysis, let us consider patient treatment data, as shown in Table 1. A patient, identified by a SSN, is diagnosed by a doctor. A patient obtains prescriptions, each of which is described by: a unique identifier, date of drawing, patient SSN and his/her age, doctor identifier, medicine name, a dose, a package capacity, and a discount the patient is eligible for.

Table 1. Example data on patient medicine prescription

prescriptionNo.	date	patientSSN	age	doctorID	medicine	dose	package	discount
1198/2011	26.04.2011	74031898333	37	3424	zinnat	0.25 g/5 ml	5 0ml	70%
1199/2011	26.04.2011	98111443657	13	3424	pulmeo	5 ml	150 ml	96.5%
3023/2011	27.04.2011	98111443657	13	9843	pulmicort	0.125 mg/ml	20 ml	70%
3024/2011	27.04.2011	98111443657	13	9843	ventolin	1.5 ml	100 ml	70%
3024/2011	27.04.2011	74031898333	37	5644	augmentin	0,6 g/5 ml	100 ml	70%
3026/2011	27.04.2011	98111443657	13	9843	ventolin	0.1 mg/ml	10 a 2 ml	70%
3027/2011	28.04.2011	34122224334	77	9843	zyrtec	1 mg/ml	75 ml	100%
3031/2011	30.04.2011	56090565958	66	9843	pulmicort	0.125 mg/ml	40 ml	100%

Typical OLAP analyses could include: (1) finding the average number of prescriptions filled by a single doctor monthly during one year period, or (2) finding the total amount of medicines prescribed by doctors working at hospital X.

However, from exploiting the sequential dependencies between data we could mine a valuable knowledge. Examples of such analyses could include: (1) finding the number of patients that were treated with medicine A after they were treated with medicine B, (2) finding the number of patients that within one year were treated at least two times with medicine A, but these treatments were separated with at least one treatment with medicine B. We argue that these and many other analyses require a new data model and query language.

3 Data Model for Sequential Time Point-Based Data

The foundation of our model includes an event and a sequence. A sequence is created from events by clustering and ordering them. Sequences and events have distinguished attributes - measures that can be analyzed in an OLAP-like manner in contexts set up by dimensions.

3.1 Building Elements of the Model

Event. We assume that an elementary data item is an *event*, whose duration is a chronon. Formally, event $e_i \in \mathbb{E}$, where $\mathbb{E}$ is the set of events, is a n-tuple $(a_{i1}, a_{i2}, ..., a_{in})$, where $a_{ij} \in dom(A_j)$. $dom(A_j)$ is the domain of attribute A_j and $dom(A_j) = \mathbb{V}$, where $\mathbb{V}$ is the set of atomic values (character string, date, number) ∪ null value. $A_j \in \mathbb{A}$, where $\mathbb{A}$ is the set of event attributes.

Attribute hierarchy. Similarly like in traditional OLAP, event attributes may have some hierarchies associated. Let $\mathbb{L} = \{L_1, L_2, ..., L_k\}$ be the set of levels in the **hierarchies** of the event attributes. Pair $\langle \mathbb{L}_{A_i}, \rhd_{A_i} \rangle$ describes a hierarchy of attribute $A_i \in \mathbb{A}$, where $\mathbb{L}_{A_i} \subseteq \mathbb{L}$ and $\rhd_{A_i}$ is a partial order on set $\mathbb{L}_{A_i}$.

Example 1. In the example from Section 2, an event represents drawing a prescription for a patient. Thus, one event is represented by one row in Table 1, i.e., $\mathbb{E} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$, where:

- e_1=(1198/2011, 26.04.2011, 74031898333, 37, 3424, zinnat, 0.25 g/5ml, 50 ml, 70%),
- e_2=(1199/2011, 26.04.2011, 98111443657, 13, 3424, pulmeo, 5 ml, 150 ml, 96.5%),
- ...
- e_8=(3031/2011, 30.04.2011, 56090565958, 66, 9843, pulmicort, 0.125 mg/ml, 40 ml, 100%).

$\mathbb{A} = \{prescriptionNo, date, patientSSN, age, doctorID, medicine, dose, package, \ discount\}$. Two attributes have the following hierarchies ($\mathbb{L} = \mathbb{L}_{A_2} \cup \mathbb{L}_{A_3}$):

- $\mathbb{L}_{A_2} = \{date, month, quarter, year\}$ and $\rhd_{A_2}$: *date* → *month* → *quarter* → *year* (with values: *26.04.2011* → *April 2011* → *Q2 2011* → *2011*),
- $\mathbb{L}_{A_3} = \{person, patient\ type\}$ and $\rhd_{A_3}$*person* → *patient type* (with values: *98111443657* → *child*, *34122224334* → *retired*).

Sequence. An ordered list of events that fulfill a given condition is called a *sequence*. The order of events in a sequence is defined by values of selected event attributes. Such attributes will further be called *ordering attributes*. A sequence is composed of the events that have the same value of another selected attribute (or attributes). Such attributes will further be called *forming attributes*. If a forming attribute has a hierarchy associated, then a selected level in the hierarchy can also be used as a forming attribute. Formally, $S_i \in \mathbb{S}$, where $\mathbb{S}$ is the set of sequences, is pair $\langle \mathbb{E}_i, \rhd \rangle$, where $\mathbb{E}_i \subseteq \mathbb{E}$ and $\rhd$ is a partial order on $\mathbb{E}$.

Creating a sequence. For the purpose of creating a sequence from events, we define operator *CreateSequence*: $\mathbb{E} \rightarrow \mathbb{S}$, with the syntax $CreateSequence(\mathbb{E}, \mathbb{F}, \mathbb{A}_o, p)$, where:

- $\mathbb{E}$ is th set of elementary events,
- $\mathbb{F}$ is the set of pairs $\langle A_i, L_j \rangle$, where $A_i \in \mathbb{A}$ is a forming attribute and $L_j \in \mathbb{L}_{A_i}$ is the level of the forming attribute A_i, or $\langle A_i, \phi \rangle$ if attribute A_i does not have a hierarchy,
- $\mathbb{A}_o$ is the set of ordering attributes, $\mathbb{A}_o \subseteq \mathbb{A}$,
- $p \in \mathbb{P}$ is a logical predicate which selects events to form sequences.

Notice that sequences are not defined statically, but their structure is dynamically constructed based on the features of analyses, for which the sequences are created.

Example 2. For the purpose of analyzing patients' treatments, all events describing prescriptions given to the same patient are included into one sequence. They are further ordered by a drawing date.

$CreateSequence(\mathbb{E}, \{\langle A_3, patient\rangle\}, \{A_2\}, null) = \{S_1, S_2, S_3, S_4\}$ where: $S_1 = \langle e_1, e_5\rangle$, $S_2 = \langle e_2, e_3, e_4, e_6\rangle$, $S_3 = \langle e_7\rangle$, $S_4 = \langle e_8\rangle$.

For the purpose of analyzing prescriptions drawn by a doctor during a year, with discounts equal to or greater than 80%, all events describing prescriptions given by the same doctor are included into a sequence. They are further ordered by a drawing date.

$CreateSequence(\mathbb{E}, \{\langle A_5, \phi\rangle \langle A_2, year\rangle\}, \{A_2\}, A_9 \geq 80\%) = \{S_1, S_2\}$ where: $S_1 = \langle e_2\rangle$, $S_2 = \langle e_7, e_8\rangle$.

Fact and measure. Any sequence which is the subject of analysis is the *fact* of analysis. Similarly as in traditional OLAP, sequences are characterized by the values of their *measure* attributes. Measure $m_i \in \mathbb{M}$, where $\mathbb{M}$ is the set of measures. $dom(m_i) = \mathbb{V}$. A measure can be either the attribute of an event or the property of the whole sequence. In order to treat measures uniformly, a measure is defined as function *ComputeMeasure* that associates an atomic value with a sequence, i.e., $ComputeMeasure : \mathbb{S} \times \mathbb{M} \rightarrow \mathbb{V}$. The syntax of the function is as follows: $ComputeMeasure(S_i, name_j, p_j)$, where:

- $S_i \in \mathbb{S}$ is a sequence, for which the values of the measure are computed,
- $name_j$ is the name of the measure,
- $p_j \in \mathbb{P}$ is an expression that computes the values of the measure for a given sequence.

Example 3. Examples of measures being event's attributes include: patient's age, medicine's doze and discount rate. Examples of measures being properties of a sequence include: duration of patient's treatment (a number of days between events: the first and last ones in sequence which describes patient's treatment) or the number of prescriptions drawn by a doctor within a day.

Dimension. A dimension sets up the context of an analysis and defines aggregation paths of facts. Let D_i denote a dimension and $\mathbb{D}$ denote the set of dimensions, thus $D_i \in \mathbb{D}$. A dimension can be either an event attribute or the property of the whole sequence. The *CreateContext* operator associates a dimension with either an event attribute or the whole sequence. It also defines a dimension hierarchy, namely the set of levels and a partial order on this set. The syntax of the operator is as follows: $CreateContext(name_{D_i}, A_{D_i}, p_{D_i}, \mathbb{H}_{D_i}) = D_i$, where:

- $name_{D_i}$ is the name of dimension D_i,
- A_{D_i} equals to $A_j \in \mathbb{A}$ if the dimension is an event attribute A_j or $A_{D_i} = \phi$ if the dimension is the property of the whole sequence,
- p_{D_i} equals to predicate $p \in \mathbb{P}$ if the dimension is the property of the whole sequence (p is an expression that computes the values of the dimension) or $p_{D_i} = \phi$ if the dimension is an event attribute,
- $\mathbb{H}_{D_i}$ is the set of hierarchies of dimension D_i, composed of pairs $\langle \mathbb{L}_{D_i}, \rhd_{D_i}\rangle$, where $\mathbb{L}_{D_i} \subseteq \mathbb{L}$ is the set of levels in the dimension hierarchy and $\rhd_{D_i}$ is a partial order on set $\mathbb{L}_{D_i}$; $\mathbb{H}_{D_i} = \phi$ if dimension D_i does not have a hierarchy.

Example 4. An example of a dimension defined by means of attributes include *patient* with hierarchy: *person* → *patient type*; the dimension is set up by an operator

$CreateContext(patient, A_3, \phi, \{\langle\{person, patient\ type\}, person \rightarrow patient\ type\rangle\})$. However, if we would like to analyze the distribution of treatment length, the dimension should be defined as a function which calculates the length of a sequence describing treatment of a single patient. In this case the dimension is defined as follows: $CreateContext(treatment\ length, \phi, for\ all\ S_i \in \mathbb{S}\ find\ Tail(S_i).A_2 - Head(S_i).A_2, \phi)$.

3.2 Operations of the Model

The operations defined in our model are classified into: (1) operations on sequences, (2) general operations, (3) operations on dimensions, and (4) analytical functions. Due to space limitations, we briefly describe the operations in this section.

Operations On Sequences – transform the structure of a single sequence and their result is another sequence. The operations include:

1. $Head(S_i)$ – removes from sequence $S_i \in \mathbb{S}$ all elements except the first one, e.g., $Head(\langle e_2, e_3, e_4, e_6\rangle) = \langle e_2\rangle$.
2. $Tail(S_i)$ – removes from sequence $S_i \in \mathbb{S}$ all elements except the last one, e.g., $Tail(\langle e_2, e_3, e_4, e_6\rangle) = \langle e_6\rangle$.
3. $Subsequence(S_i, m, n)$ – removes from sequence $S_i \in \mathbb{S}$ all elements prior to the element at position m and all elements following element at position n, e.g., $Subsequence(\langle e_2, e_3, e_4, e_6\rangle, 2, 3) = \langle e_3, e_4\rangle$.
4. $Split(S_i, expression)$ – splits sequence $S_i \in \mathbb{S}$ into the set of new sequences based on *expression*; each element of the original sequence belongs to only one of the resulting sequences, e.g., $Split(\langle e_2, e_3, e_4, e_6\rangle,$"the same values of A_5") $= \{\langle e_2\rangle, \langle e_3, e_4, e_6\rangle\}$ (the original sequence describes the whole treatment of a patient regardless of doctors, whereas the resulting set of sequences represents the treatments of the same patient but now each treatment is conducted by one doctor).
5. $Combine(\mathrm{S})$ – creates a new sequence from elements of all sequences in $\mathrm{S} \subseteq \mathbb{S}$ given as parameters; the elements in a new sequence are ordered by the values of ordering attributes of the original sequences, e.g., $Combine(\{\langle e_2\rangle, \langle e_3, e_4, e_6\rangle\}) = \langle e_2, e_3, e_4, e_6\rangle$ (the original sequences describe treatments of the same patient but by two different doctors and the resulting sequence represents the whole treatment of the patient).

General Operations – allow to manipulate sets of sequences.

1. $Pin(\mathrm{S}, expression)$ – filters sequences $\mathrm{S} \subseteq \mathbb{S}$ that fulfill a given expression, e.g., $Pin(\mathrm{S}, length(S_i \in \mathrm{S}) > 3)$ (rejects all sequences that consist of less than four events).
2. $Select(\mathrm{S}, expression)$ – removes from $\mathrm{S} \subseteq \mathbb{S}$ events that do not fulfill a given expression, e.g., $Select(\mathrm{S}, A_6 = zinnat)$ (removes from sequences all events that concern prescriptions other than "zinnat").
3. $GroupBy(\mathrm{S}, expression \mid D_i)$ – assigns sequences from $\mathrm{S} \subseteq \mathbb{S}$ to groups according to the results of a given grouping *expression* (case A) or to dimensions in $D_i \in \mathbb{D}$ (case B). Sequences with the same value of the grouping expression or dimension value belong to one group. The result of the operation is set $\mathbb{G}$ of pairs $\langle value, \mathrm{S}_i\rangle$,

where *value* is a given value of grouping expression and $\mathrm{S}_i \subseteq \mathbb{S}$ is the set of sequences with the same value of a grouping expression (case A), or set of pairs $\langle value(D_i), \mathrm{S}_i \rangle$, where $value(D_i)$ is the value of dimension D_i (case B).

4. Set operations: union $\cup$, difference $\setminus$, and intersection $\cap$ – they are standard set operations that produce a new set of sequences, e.g., $\mathrm{S}_1 \cup \mathrm{S}_2$.

Operations on Dimensions – allow to navigate in the hierarchy of a given dimension. The operations include:

1. $LevelUp(\mathrm{D}, \mathrm{S})$ navigates one level up in the hierarchy of dimensions in D (where $\mathrm{D} \subseteq \mathbb{D}$) for all sequences in $\mathrm{S} \subseteq \mathbb{S}$.
   An example of this operation main include changing the levels of attribute A_2 - from *date* to *month* and A_3 - from *person* to *patient type*: $LevelUp(\{date, patient\}, \{S_1\}) = \{S_1'\}$, where
   - $S_1 = \langle e_1, e_5 \rangle$ (e_1=(1198/2011, 26.04.2011, 74031898333, ..., 50 ml, 70%), e_5=(3024/2011, 27.04.2011, 74031898333, ..., 100 ml, 70%)),
   - $S_1' = \langle e_1', e_5' \rangle$ (e_1'=(1198/2011, April 2011, regular, ..., 50 ml, 70%), e_5'=(3024/2011, April 2011, regular, ..., 100 ml, 70%)).
2. $LevelDown(\mathrm{D}, \mathrm{S})$ navigates one level down in the hierarchy of dimensions in D (where $\mathrm{D} \subseteq \mathbb{D}$) for all sequences in $\mathrm{S} \subseteq \mathbb{S}$.
   An example of this operation may include changing the level of attribute A_3 from *patient type* to *person*: $LevelDown(\{patient\}, \{S_1'\}) = \{S_1''\}$, where
   - $S_1'' = \langle e_1'', e_5'' \rangle$ (e_1''=(1198/2011, April 2011, 74031898333, ..., 50 ml, 70%), e_5''=(3024/2011, April 2011, 74031898333, ..., 100 ml, 70%)).

Analytical Functions – compute aggregates of measures. They include OLAP-like functions *Count*, *Sum*, *Avg*, *Min*, and *Max*. For example, in order to compute the number of sequences formed with attribute A_3 (*patientSSN*) equal to 98111443657 the following expression is used: $Count(Pin(\mathbb{S}, A_3 = 98111443657))$.

Example 5. In order to illustrate the application of our model, let us consider two simple analyses.

Find the number of patients who were treated at least two times with the same medicine in 2000, and in between they were prescribed a different medicine. The implementation of this query using the presented model is as follows:

1. $\mathbb{S} = CreateSequence(\mathbb{E}, \{\langle A_2, year \rangle, \langle A_3, person \rangle\}, \{A_2\},$
   A_2 *between* 1.1.2010 *and* 31.12.2010)
2. $\mathbb{S}' = Pin(\mathbb{S}, e_i.A_6 = e_{i+2}.A_6 \; and \; e_i.A_6 ! = e_{i+1}.A_6)$
3. $Count(\mathbb{S}')$.

Find distribution of treatment lengths. The implementation of this query is as follows:

1. $\mathbb{S} = CreateSequence(\mathbb{E}, \{\langle A_3, person \rangle\}, \{A_2\}, null)$
2. $D_1 = CreateContext(treatment\ length, \phi, for\ all\ S_i \in \mathbb{S}\ find\ Tail(S_i).A_2 - Head(S_i).A_2, \phi)$
3. $\mathbb{G} = GroupBy(\mathbb{S}, D_1)$
4. $Count(\mathbb{G})$.

4 Related Work

The research and technological areas related to processing sequential data include: (1) complex event processing (CEP) over data streams and (2) OLAP. The CEP technology has been developed for the purpose of continuous analysis of data streams for the purpose of detecting patterns, outliers, and generating alerts, e.g., [4,9]. On the contrary, the OLAP technology [5] has been developed for the purpose of analyzing huge amounts of data organized in relations but it is unable to exploit the sequential nature of data. In this respect, *Stream Cube* [13] and *E-Cube* [14] implement OLAP on data streams. Their main focus is on providing tools for OLAP analysis within a given time window of constantly arriving streams of data.

Another research problem is storage and analysis of data whose inherent feature is an order. These problems have been researched since several years with respect to storage, e.g., [20,17] and query processing, e.g., [19,18]. In [20] sequences are modeled by an enhanced abstract data type, in an object-relational model, whereas in [17] sequences are modeled as sorted relations. The query languages proposed in [19,18] allow typical OLTP selects on sequences but do not support OLAP analyzes. Further extension towards sequence storage and analysis have been made in [6] that proposes a general concept of a RFID warehouse. Unfortunately, no in-dept discussion on RFID data storage and analysis has been provided.

[7,8,15] focus on storage and analysis of time point-based sequential data. [15] propose the set of operators for a query language for the purpose of analyzing patterns. [7,8] focus on an algorithm for supporting ranking pattern-based aggregate queries and on a graphical user interface. The drawback of these approaches is that they are based on relational data model and storage for sequential data.

[10,11] address interval-based sequential data, generated by RFID devices. The authors focus on reducing the size of such data. They propose techniques for constructing RFID cuboids and computing higher level cuboids from lower level ones. For storing RFID data and their sequential orders the authors propose to use three tables, called Info, Stay, and Map.

Our approach differs from the related ones as follows. First, unlike [13,14], we focus on analyzing sequential data stored in a data warehouse. Second, unlike [20,17,18], we concentrate on an OLAP-like analysis of sequential data. Third, unlike [10,11], we focus on analyzing time point-based sequential data. Finally, unlike [7,8,15], we propose a new formal model for sequential data, since in our opinion a relational-like data model is not sufficient for the support of fully functional analysis.

5 Conclusions and Future Work

In this paper we proposed a formal model for representing and analyzing time point-based sequential data, based on the notion of an *event* and a *sequence*, where sequences are dynamically created from events. In order to support OLAP-like analyses of sequences, we associate with events and sequences attributes representing *measures*. Next, we analyze sequences in contexts defined by *dimensions*. Dimensions can have hierarchies, like in traditional OLAP. Sequence analysis is performed by four classes of operations, i.e., operations on sequences, on dimensions, general operations, and analytical functions. To the best of our knowledge, this is the first comprehensive formal model and query language for this class of data. Based on the model, we are currently developing a query processor and results visualizer. Future work will focus on internal mechanisms, like query optimization and data structures.

OLAP-like analysis of interval-based data requires even more advanced data model and operators, e.g., computing aggregates on intervals requires additional semantics of aggregate functions; intervals sort criteria may include begin time, end time, or midpoint time, resulting in different interval sequences; creating and comparing sequences of intervals requires well defined operators. For this reason, we start our research with a simpler problem and in the future we plan to extend our findings towards the interval-based model.

References

1. Octopus card, http://hong-kong-travel.org/Octopus/ (retrieved March 30, 2012)
2. Smart card alliance, http://www.smartcardalliance.org (retrieved March 30, 2012)
3. Bębel, B., Krzyżagórski, P., Kujawa, M., Morzy, M., Morzy, T.: Formal model for sequential olap. In: Information Technology and its Applications. NAKOM (2011) ISBN 978-83-89529-82-4
4. Buchmann, A.P., Koldehofe, B.: Complex event processing. IT - Information Technology 51(5), 241–242 (2009)
5. Chaudhuri, S., Dayal, U., Narasayya, V.: An overview of business intelligence technology. Communications of the ACM 54(8), 88–98 (2011)
6. Chawathe, S.S., Krishnamurthy, V., Ramachandran, S., Sarma, S.: Managing rfid data. In: Proc. of Int. Conf. on Very Large Data Bases (VLDB), pp. 1189–1195 (2004)
7. Chui, C.K., Kao, B., Lo, E., Cheung, D.: S-olap: an olap system for analyzing sequence data. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 1131–1134. ACM (2010)
8. Chui, C.K., Lo, E., Kao, B., Ho, W.-S.: Supporting ranking pattern-based aggregate queries in sequence data cubes. In: Proc. of ACM Conf. on Information and Knowledge Management (CIKM), pp. 997–1006. ACM (2009)
9. Demers, A.J., Gehrke, J., Panda, B., Riedewald, M., Sharma, V., White, W.M.: Cayuga: A general purpose event monitoring system. In: CIDR, pp. 412–422 (2007)
10. Gonzalez, H., Han, J., Li, X.: FlowCube: constructing rfid flowcubes for multi-dimensional analysis of commodity flows. In: Proc. of Int. Conf. on Very Large Data Bases (VLDB), pp. 834–845 (2006)
11. Gonzalez, H., Han, J., Li, X., Klabjan, D.: Warehousing and analyzing massive rfid data sets. In: Proc. of Int. Conf. on Data Engineering (ICDE), p. 83. IEEE (2006)
12. Gorawski, M.: Multiversion Spatio-temporal Telemetric Data Warehouse. In: Grundspenkis, J., Kirikova, M., Manolopoulos, Y., Novickis, L. (eds.) ADBIS 2009. LNCS, vol. 5968, pp. 63–70. Springer, Heidelberg (2010)
13. Han, J., Chen, Y., Dong, G., Pei, J., Wah, B.W., Wang, J., Cai, Y.D.: Stream cube: An architecture for multi-dimensional analysis of data streams. Distributed and Parallel Databases 18(2), 173–197 (2005)
14. Liu, M., Rundensteiner, E., Greenfield, K., Gupta, C., Wang, S., Ari, I., Mehta, A.: E-cube: multi-dimensional event sequence analysis using hierarchical pattern query sharing. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 889–900. ACM (2011)

15. Lo, E., Kao, B., Ho, W.-S., Lee, S.D., Chui, C.K., Cheung, D.W.: Olap on sequence data. In: Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp. 649–660. ACM (2008)
16. Mörchen, F.: Unsupervised pattern mining from symbolic temporal data. SIGKDD Explor. Newsl. 9(1), 41–55 (2007)
17. Ramakrishnan, R., Donjerkovic, D., Ranganathan, A., Beyer, K.S., Krishnaprasad, M.: Srql: Sorted relational query language. In: Proc. of Int. Conf. on Scientific and Statistical Database Management (SSDBM), pp. 84–95. IEEE (1998)
18. Sadri, R., Zaniolo, C., Zarkesh, A., Adibi, J.: Optimization of sequence queries in database systems. In: Proc. of ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), pp. 71–81. ACM (2001)
19. Seshadri, P., Livny, M., Ramakrishnan, R.: Sequence query processing. SIGMOD Record 23(2) (1994)
20. Seshadri, P., Livny, M., Ramakrishnan, R.: The design and implementation of a sequence database system. In: Proc. of Int. Conf. on Very Large Data Bases (VLDB), pp. 99–110. Morgan Kaufmann Publishers Inc. (1996)

Multi-dimensional Navigation Modeling Using BI Analysis Graphs*

Thomas Neuböck[1], Bernd Neumayr[2],
Thomas Rossgatterer[1], Stefan Anderlik[2], and Michael Schrefl[2]

[1] Solvistas GmbH, Graben 18, 4020 Linz, Austria
{thomas.neuboeck,thomas.rossgatterer}@solvistas.at
[2] Department of Business Informatics – Data & Knowledge Engineering,
Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria
{neumayr,anderlik,schrefl}@dke.uni-linz.ac.at
http://www.solvistas.at
http://www.dke.jku.at

Abstract. To solve analysis tasks in business intelligence, business analysts frequently apply a step-by-step approach. Using their expert knowledge they navigate between different measures, also referred to as business ratios or key performance indicators, at different levels of detail and focus on different aspects or parts of their organization. In this paper we introduce *BI Analysis Graphs* to document and analyze these navigation steps. We further introduce BI Analysis Graph *Templates* to model and re-use recurrent navigation patterns. We describe reasoning tasks over BI Analysis Graph Templates and sketch how they are implemented in our proof-of-concept prototype. BI Analysis Graph Templates may serve as formal foundation for interactive dashboards and guided analytics in business intelligence applications.

Keywords: Data warehouse, Decision support, Interactive data exploration, Knowledge modeling, Conceptual modeling.

1 Introduction

Business analysts use OLAP tools, interactive dashboards and other Business Intelligence (BI) systems to gain insights into an object of analysis and to inform the decision-making process. In a stepwise manner, business analysts navigate through multi-dimensional data: they look at different measures, ratios and performance indicators, for different areas or parts of the object of analysis, and at different levels of detail. This is what we call *multi-dimensional navigation*. In our project* we are confronted with the need of our project partners from health insurance industry to better support business analysts in sharing their

* This work is funded by the Austrian Ministry of Transport, Innovation, and Technology in program FIT-IT Semantic Systems and Services under grant FFG-829594 (*Semantic Cockpit:* an ontology-driven, interactive business intelligence tool for comparative data analysis).

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 162–171, 2012.

insights with others and in documenting and re-using their analyses. We believe that in order to understand the insights of business analysts and to externalize their knowledge one needs to understand their navigation steps. In this sense we state *'Navigation is Knowledge'*.

Navigation modeling has received considerable attention, especially in model-driven web engineering [1]. Previous work has studied modeling OLAP navigation [2,3] and describing analytical sessions [4], i.e., sequences of related OLAP queries. There is, however, still a lack for a conceptual formalism that helps to share, understand and re-use navigational knowledge of business analysts. The contribution of this paper is to introduce the core of such a formalism.

In this paper we introduce *BI Analysis Graphs* for documenting and analyzing the steps an analyst takes when solving an analysis task. A BI Analysis Graph is a directed acyclic graph and consists of *analysis situations* as vertices and *analysis steps* labeled by *navigation operations* as directed edges.

An *analysis situation* represents a query against the BI system. An analysis situation is defined by the following properties: a multi-dimensional *point*, a single *complex derived measure* (a notion taken from [5]), an optional *qualification* and an optional *grouping granularity*.

An *analysis step* represents the navigation of an analyst from a *source* analysis situation to a *target* analysis situation. In order to facilitate understanding the navigational behavior of analysts we decompose it to *atomic* analysis steps. An atomic step connects a source analysis situation with a target analysis situation which differ in only one of their properties. The difference between source and target is indicated by a *navigation operation* which is defined by a navigation operator and a parameter. We revisit and introduce different *navigation operators* which define different types of atomic steps. Source and target situation either differ in a coordinate of the point (navigation operators *moveDown*, *moveUp*, *moveAside*), in the measure (*refocus*), in a coordinate of the grouping granularity (*drillDown*, *rollUp*, *split*, *merge*), or in the qualification (*narrow*, *broaden*).

We further introduce *BI Analysis Graph Templates* for modeling and re-using recurrent navigation patterns. A BI Analysis Graph Template is a BI Analysis Graph with free variables. Free variables are bound to concrete values during analysis. A free variable may either stand for a measure, a qualification, a coordinate of a point, or a coordinate of a grouping granularity. Since we do not prescribe which parts of a template are to be constant and which are to be variable, one may use templates in a very versatile manner. The main reasoning tasks for BI Analysis Graph Templates are to check their consistency and to find possible instantiations. Both reasoning tasks are implemented in a Datalog-based proof-of-concept prototype.

The remainder of the paper is organized as follows. In Section 2 we revisit some constructs from multi-dimensional modeling as prerequisites for defining analysis situations in Section 3 and analysis steps in Section 4. BI Analysis Graphs are discussed in Section 5, followed by BI Analysis Graph Templates and reasoning in Section 6. In Section 7 we give an overview of related work and conclude the paper in Section 8 with an outlook on future work.

2 Multi-dimensional Content Model

In this section we clarify how we understand some basic notions from multi-dimensional modeling: dimensions, complex-derived measures, and predicates. Thereby, we introduce our multi-dimensional content model, which we understand to be at the 'semantic layer' above the data warehouse. We abstract away from the physical and logical organization of data in cubes or tables, and are instead only interested in (complex-derived) measures, their applicability on points in the multi-dimensional space, and their specialization by predicates.

A *dimension* is a rooted directed acyclic graph of nodes describing a roll-up hierarchy, with $\top$ as root node. If a node o rolls up directly or indirectly or is equivalent to node o' we write $o \uparrow o'$. Dimensions provide for *summarizability* by being homogeneously leveled. Each node o of a dimension belongs to one level of this dimension, the level of node o is denoted as $l(o)$. Each dimension consists of a set of levels which are arranged in a lattice with a bottom or base level, $\underline{\bot}$, and a top level, $\overline{\top}$. This partial order of levels describes the rollup hierarchy at the schema level. If a level l rolls up to a level l', denoted as $l \uparrow l'$, then each node at l rolls up to exactly one node at l'. For simplicity we assume, for the rest of the paper, a fixed number of n dimensions and a fixed ordering of these dimensions and refer to a dimension by an integer i (with $0 < i \leq n$).

Let D_i be the nodes of a dimension i and l a level of the dimension. We denote by D_i^l the set of nodes of dimension i at level l. The set of nodes at the base level are denoted by $D_i^{\underline{\bot}}$. The top level only consists of the top node $D_i^{\overline{\top}} = \{\top\}$. Note that each dimension has its own top level, $\overline{\top}_i$, own base level, $\underline{\bot}_i$, and own top node, $\top_i$; we omit the subscripted dimension index if the dimension is given by the context. The part of dimension i which is rooted in node $\hat{o}$, denoted as $D_{i/\hat{o}}$, is the set of nodes that roll up to $\hat{o}$, i.e. $D_{i/\hat{o}} = \{o' \in D_i \mid o' \uparrow \hat{o}\}$. The set of nodes at level l that roll up to node $\hat{o}$ is denoted as $D_{i/\hat{o}}^l$. The set of nodes that roll up to node $\hat{o}$ at levels from a level $l^\vartriangle$ (the lower bound) to a level $l^\triangledown$ (the upper bound) are denoted as $D_{i/\hat{o}}^{l^\vartriangle..l^\triangledown}$, i.e., $D_{i/\hat{o}}^{l^\vartriangle..l^\triangledown} = \{o' \in D_i \mid o' \uparrow \hat{o} \wedge l^\vartriangle \uparrow l(o') \uparrow l^\triangledown\}$. The set of nodes of a dimension i, D_i, can thus also be denoted as $D_{i/\top}^{\underline{\bot}..\overline{\top}}$.

Example 1 (Dimensions, Levels and Nodes). In our simplified example from health insurance industry we analyze drug prescriptions and doctor contacts of insurants along the dimensions *location* ($i = 1$) and *time* ($i = 2$). Dimension *location* has levels *country*, *province*, *city* such that $\underline{\bot}_1 = \mathit{city} \uparrow \mathit{province} \uparrow \mathit{country} \uparrow \overline{\top}_1$. Dimension *time* has levels $\underline{\bot}_2 = \mathit{day} \uparrow \mathit{month} \uparrow \mathit{quarter} \uparrow \mathit{year} \uparrow \overline{\top}_2$. In subsequent examples we only consider the following nodes: $D_{1/\top}^{\mathit{country}} = \{\mathit{Austria}\}$, $D_{1/\mathit{Austria}}^{\mathit{province}} = \{\mathit{UpperAustria}, \mathit{LowerAustria}\}$, $D_{2/\top}^{\mathit{year}} = \{\mathit{2012}\}$, $D_{2/\mathit{2012}}^{\mathit{quarter}} = \{\mathit{2012Q1}, \mathit{2012Q2}, \mathit{2012Q3}, \mathit{2012Q4}\}$.

A complex derived measure (in the following simply referred to as *measure*) consists of a signature and an implementation. The *signature* of a measure is given by its name and its *multi-dimensional domain.* A measure may be applied on each point $\langle o_1, \ldots, o_n \rangle$ in its multi-dimensional domain $D_{1/\hat{o}_1}^{l_1^\vartriangle..l_1^\triangledown} \times \cdots \times D_{n/\hat{o}_n}^{l_n^\vartriangle..l_n^\triangledown}$.

The *measurement instruction* (or *measure implementation*) defines how to derive or calculate —for a given point— a measure value from the data in the data warehouse. In this paper we do not further discuss measurement instructions and treat measures as black-box.

Measures may be specialized by giving a predicate as qualification. Then, only base facts fulfilling the predicate are considered for calculating the measure value. Predicates are arranged in a *subsumption hierarchy.* We write $q \sqsubseteq q'$ to denote that predicate q' subsumes predicate q.

Example 2 (Measures and Predicates). Measure *drugCosts* is defined as the total costs of drug prescriptions. It is defined for multi-dimensional domain $D_{1/\top}^{\perp..\top} \times D_{2/\top}^{\perp..\top}$. Measure *avgNrOfAnnualContacts* is defined as the average number of doctor contacts per insurant and year and may be applied on points in the multi-dimensional domain $D_{1/\top}^{\perp..\top} \times D_{2/\top}^{year..\top}$. Note that this measure is derived from a measure aggregated to level *year*, since that it has *year* as minimal granularity in dimension *time*. Predicate *dm2* is true for each fact which is about insurants suffering from diabetes mellitus type 2 (DM2) and predicate *dm2oad* is true for each fact which concerns both insurants suffering from DM2 and oral antidiabetic drugs (OAD). It holds that *dm2oad* $\sqsubseteq$ *dm2*.

3 Analysis Situations

In this section we exemplify and define analysis situations. Analysis situations represent queries against the data warehouse and are used in subsequent sections to define navigation operations and analysis graphs.

An *analysis situation* $A = (\langle o_1, ..., o_n\rangle, m, q, \langle g_1, ..., g_n\rangle)$ consists of a point $\langle o_1, \ldots, o_n\rangle$, a measure m, an optional qualification q and a grouping granularity $\langle g_1, \ldots, g_n\rangle$. When evaluated against a data warehouse it results in a fact consisting of point $\langle o_1, \ldots, o_n\rangle$ and a *resulting cube.* The grouping granularity indicates dimensionality and granularity of the resulting cube. Each coordinate g_i of the grouping granularity is either a level of dimension i or is unspecified, denoted as $g_i = \tau$. A coordinate $g_i = \tau$ means that dimension i is not considered as dimension in the resulting cube. This also means that the *value granularity*, i.e., the granularity of measure values in the resulting cube, not only depends on the grouping granularity but also on the granularity of the point. We use this value granularity in order to check if an analysis situation complies with the multi-dimensional domain of its measure. The value granularity $\langle l_1, \ldots, l_n\rangle$ is defined as (for $i = 1..n$): $l_i = g_i$, if $g_i \neq \tau$, and $l_i = l(o_i)$, if $g_i = \tau$.

Let $D_{1/\hat{o}_1}^{l_1^{\vartriangle}..l_1^{\triangledown}} \times \cdots \times D_{n/\hat{o}_n}^{l_n^{\vartriangle}..l_n^{\triangledown}}$ be the multi-dimensional domain of measure m, then an analysis situation is *consistent*, denoted as $\gamma(\langle o_1, ..., o_n\rangle, m, q, \langle g_1, ..., g_n\rangle)$, if in each dimension ($i = 1..n$) the node o_i rolls up to $\hat{o}_i$ and the minimal granularity $l_i^{\vartriangle}$ rolls up to the value granularity l_i (as defined above) which in turn rolls up to the maximal granularity $l_i^{\triangledown}$ of the measure and to the granularity $l(o_i)$ of the point, i.e., $\gamma(\langle o_1, ..., o_n\rangle, m, q, \langle g_1, ..., g_n\rangle) \overset{\text{def}}{=} \bigwedge_{i=1}^{n} o_i \uparrow \hat{o}_i \wedge l_i^{\vartriangle} \uparrow l_i \wedge l_i \uparrow l_i^{\triangledown} \wedge l_i \uparrow l(o_i)$.

Example 3 (Analysis Situation). Analysis situation *drug costs for DM2 patients in Upper Austria in the year 2012 grouped by quarter* is denoted as (also see A_4 in upper part of Figure 2): $(\langle \mathit{UpperAustria}, 2012\rangle, \mathit{drugCosts}, \mathit{dm2}, \langle \tau,\ \mathit{quarter}\rangle)$. Note that the grouping granularity is unspecified for dimension *location* ($g_1 = \tau$). When evaluating this analysis situation one gets a fact with a one-dimensional resulting cube, having only dimension *time* (as depicted in the lower part of Figure 2). Now the granularity of values in this result cube (i.e., the value granularity) also depends on the level of the *location*-coordinate of the point, which is *UpperAustria*, which is at level *province*. The value granularity is thus $\langle \mathit{province}, \mathit{quarter}\rangle$ which is between minimal and maximal granularity of measure *drugCosts*.

4 Analysis Steps

In this section we look at the navigation from one situation to another, referred to as *analysis step*. For better understandability of analysts' behavior we only allow atomic analysis steps (with complex steps being decomposed to atomic ones). An atomic analysis step connects two analysis situations which only differ in a single property. We introduce different navigation operators used in BI Analysis Graphs to describe the difference between two situations.

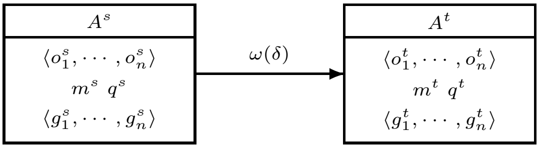

A^s
$\langle o_1^s, \cdots, o_n^s\rangle$
$m^s\ q^s$
$\langle g_1^s, \cdots, g_n^s\rangle$
$\omega(\delta)$
A^t
$\langle o_1^t, \cdots, o_n^t\rangle$
$m^t\ q^t$
$\langle g_1^t, \cdots, g_n^t\rangle$

Fig. 1. Analysis Step

An atomic analysis step connects a source analysis situation A^s and a target analysis situation A^t (as depicted in Figure 1). Such a step is atomic in that the two situations may only differ in a single property. The difference between the two situations (the interesting difference, the reason for navigation) is unambiguously described by navigation operation $\omega(\delta)$, consisting of a navigation operator ω and a parameter δ.

We now introduce, in Table 1, navigation operators to describe atomic analysis steps. Each operator has a signature, given by its name (e.g., *moveDown*) and its formal parameter (e.g., $i : o$). The parameter variables are implicitly typed (i is an integer value identifying a dimension, o is a node, g is a level or τ, m is a measure, and q is a predicate). The semantics of each operator is defined in table column "Condition". An analysis step from situation A^s to situation A^t described by navigation operation $\omega(\delta)$ is *consistent* if both situations are consistent, i.e., $\gamma(A^s)$ and $\gamma(A^t)$ hold, and the condition indicated for operator ω holds between source situation A^s, target situation A^t and parameter δ and all properties not mentioned in the condition are the same for A^s and A^t.

Table 1. Navigation Operators

Operation	Condition	Operation	Condition
moveDown$(i : o)$	$o = o_i^t \wedge o \neq o_i^s \wedge o \uparrow o_i^s$	*split*$(i : g)$	$g = g_i^t \wedge g_i^s = \tau \wedge g \neq \tau$
moveUp$(i : o)$	$o = o_i^t \wedge o \neq o_i^s \wedge o_i^s \uparrow o$	*merge*(i)	$g_i^s \neq \tau \wedge g_i^t = \tau$
moveAside$(i : o)$	$o = o_i^t \wedge o \neq o_i^s \wedge l(o_i^s) = l(o)$	*refocus*(m)	$m = m^t \wedge m \neq m^s$
drillDown$(i : g)$	$g = g_i^t \wedge g \neq g_i^s \wedge g \uparrow g_i^s$	*narrow*(q)	$q = q^t \wedge q \neq q^s \wedge q \sqsubseteq q^s$
rollUp$(i : g)$	$g = g_i^t \wedge g \neq g_i^s \wedge g_i^s \uparrow g$	*broaden*(q)	$q = q^t \wedge q \neq q^s \wedge q^s \sqsubseteq q$

By means of *moveDown*-, *moveUp*-, and *moveAside*-operations a user navigates from one point to another by changing one coordinate; *moveDown* and *moveUp* navigate downwards or upwards the roll-up hierarchy and *moveAside* changes to a node at the same level. *drillDown* and *rollUp* change one coordinate of the grouping granularity; *drillDown* moves to a finer level and *rollUp* to coarser level. *split* introduces a dimension at a given level to the grouping granularity and *merge* removes a dimension. *refocus* switches to another measure. *narrow* and *broaden* change the qualification to a more specialized or to a more generalized predicate. Examples are given in the next section.

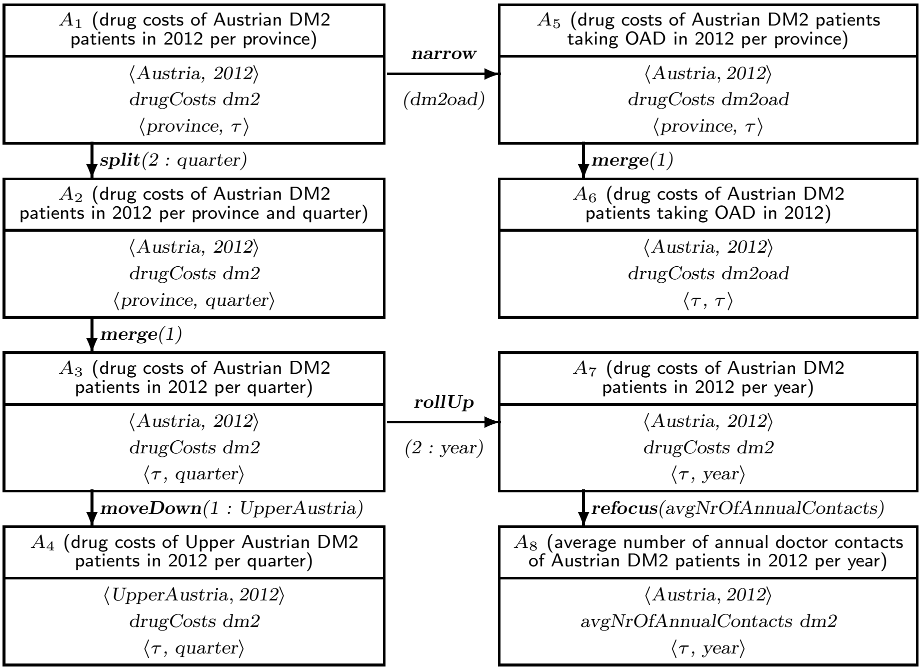
5 BI Analysis Graphs

Building on analysis situations and navigation operations introduced in previous sections we can now introduce and exemplify BI Analysis Graphs. A BI Analysis Graph documents the steps a business analyst takes in order to solve an analysis task. It can be used to analyze and reproduce past analysis sessions.

A *BI Analysis Graph* consists of analysis situations as vertices and analysis steps labeled by navigation operations as directed edges. An analysis graph is *consistent* if every vertex is a consistent analysis situation and every edge is a consistent analysis step.

Beside documentation assistance and reproduction of analysis sessions, BI Analysis Graphs also provide a tool support basis for developing analyses. Starting in an analysis situation (source situation) the navigation operators offer the next analysis steps a business analyst can choose, i.e. the analysis graph assists the analyst in determining the next analysis situation (target situation). In this way business analysts are supported in their step-by-step navigation through their systems of measurements and key performance indicators.

Example 4 (BI Analysis Graph). The upper half of Figure 2 shows a BI analysis graph, documenting the solution of an analysis task (simplified for illustrative purposes). Fictitious results of analysis situations are shown in the lower half of Figure 2. Suppose a business analyst has to analyze Austrian DM2 patients in the year 2012. To solve the task she looks at drug costs at different granularities (analysis situations A_1, A_2, A_3, and A_7), she moves down to patients in Upper Austria (analysis situation A_4), she inspects a specific group of DM2 patients (analysis situations A_5 and A_6), i.e., DM2 patients taking oral antidiabetic drugs, and/or she


A_1 (drug costs of Austrian DM2 patients in 2012 per province)
⟨Austria, 2012⟩
drugCosts dm2
⟨province, τ⟩
narrow
(dm2oad)
A_5 (drug costs of Austrian DM2 patients taking OAD in 2012 per province)
⟨Austria, 2012⟩
drugCosts dm2oad
⟨province, τ⟩
split(2 : quarter)
merge(1)
A_2 (drug costs of Austrian DM2 patients in 2012 per province and quarter)
⟨Austria, 2012⟩
drugCosts dm2
⟨province, quarter⟩
A_6 (drug costs of Austrian DM2 patients taking OAD in 2012)
⟨Austria, 2012⟩
drugCosts dm2oad
⟨τ, τ⟩
merge(1)
A_3 (drug costs of Austrian DM2 patients in 2012 per quarter)
⟨Austria, 2012⟩
drugCosts dm2
⟨τ, quarter⟩
rollUp
(2 : year)
A_7 (drug costs of Austrian DM2 patients in 2012 per year)
⟨Austria, 2012⟩
drugCosts dm2
⟨τ, year⟩
moveDown(1 : UpperAustria)
refocus(avgNrOfAnnualContacts)
A_4 (drug costs of Upper Austrian DM2 patients in 2012 per quarter)
⟨UpperAustria, 2012⟩
drugCosts dm2
⟨τ, quarter⟩
A_8 (average number of annual doctor contacts of Austrian DM2 patients in 2012 per year)
⟨Austria, 2012⟩
avgNrOfAnnualContacts dm2
⟨τ, year⟩

A_1: **location:** Austria, **time:** 2012

location:province	drugCosts dm2
UpperAustria	€ 900,000
LowerAustria	€ 1,050,000

A_5: **location:** Austria, **time:** 2012

location:province	drugCosts dm2oad
UpperAustria	€ 600,000
LowerAustria	€ 700,000

A_2: **location:** Austria, **time:** 2012

location:province	time:quarter	drugCosts dm2
UpperAustria	2012Q1	€ 300,000
UpperAustria	2012Q2	€ 200,000
UpperAustria	2012Q3	€ 150,000
UpperAustria	2012Q4	€ 250,000
LowerAustria	2012Q1	€ 350,000
...	...	...

A_6: **location:** Austria, **time:** 2012

drugCosts dm2oad
€ 1,300,000

A_3: **location:** Austria, **time:** 2012

time:quarter	drugCosts dm2
2012Q1	€ 650,000
2012Q2	€ 400,000
2012Q3	€ 450,000
2012Q4	€ 450,000

A_7: **location:** Austria, **time:** 2012

time:year	drugCosts dm2
2012	€ 1,950,000

A_4: **location:** UpperAustria, **time:** 2012

time:quarter	drugCosts dm2
2012Q1	€ 300,000
2012Q2	€ 200,000
2012Q3	€ 150,000
2012Q4	€ 250,000

A_8: **location:** Austria, **time:** 2012

time:year	avgNrOfAnnualContacts dm2
2012	14.3

Fig. 2. A BI Analysis Graph (top) and illustrative and fictitious results (bottom)

changes to another measure (analysis situation A_8), i.e., average number of annual doctor contacts per insurant. In each analysis step the navigation operation (e.g., *split(2 : quarter)*) is an explicit part of the knowledge of the business analyst, this emphasizes our introductory slogan *'Navigation is Knowledge'*.

6 BI Analysis Graph Templates

Building on analysis graphs we will now show how to represent re-usable and recurrent parts of analyses as BI Analysis Graph Templates. We will further discuss basic reasoning support and sketch its implementation in a proof-of-concept prototype.

In Figure 2 we demonstrated an analysis graph G with fixed objects (fixed measures, points, granularities, and predicates). Now we introduce BI Analysis Graphs with free variables. An analysis graph with free variables is called BI Analysis Graph Template. A BI Analysis Graph Template is instantiated by binding all free variables to concrete values. A business analyst can (re-)use such templates to analyze various analysis tasks. We regard a *BI Analysis Graph Template* $\mathcal{G}$ as a collection of analysis situations which are connected via navigation operations in the sense of the previous sections. Analysis situations and navigation operations have constants and free variables at the syntax level.

Example 5 (BI Analysis Graph Template). Suppose a business analyst wants to analyze Austrian DM2 patients. The analysis graph template $\mathcal{G}$ in Figure 3 shows a constant point $\langle \mathit{Austria}, \mathit{2012} \rangle$ and a constant qualification *dm2* that restrict the analysis to Austrian DM2 patients in the year 2012. We also have free variables prefixed by "*?*". The variables *?m* and *?g* state that the analyst should or can watch all available measures (*drugCosts*, *avgNrOfAnnualContacts*) at various location granularities (*country*, *province*). The analyst can move down to subnodes of *Austria* by binding variable *?o* to *UpperAustria* or *LowerAustria*, and she or he can further narrow the result to special groups of DM2 patients, e.g. variable *?q* can be bound to *dm2oad*.

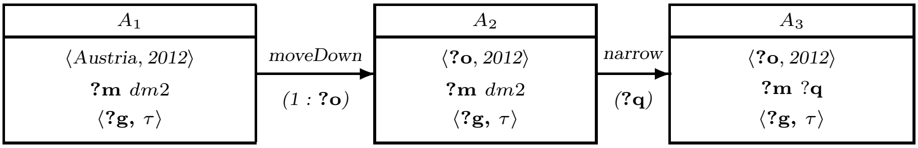


Fig. 3. BI Analysis Graph Template $\mathcal{G}$

We identified two important reasoning tasks. First, when modeling a BI Analysis Graph Template, the reasoner assists the modeler with an automatic consistency check. Second, when instantiating a template, the reasoner assists the analyst in finding possible variable bindings. In order to automatize these two reasoning tasks we regard a BI Analysis Graph Template as a first-order logic formula with free variables. For example, analysis graph template $\mathcal{G}$ in Figure 3 corresponds to the following formula:

$$\begin{array}{llllll}
\gamma(& \langle \mathit{Austria},\ \mathit{2012}\rangle, & \mathbf{?m}, & \mathit{dm2}, & \langle \mathbf{?g},\ \tau\rangle &) \quad \wedge \\
\gamma(& \langle \mathbf{?o},\ \mathit{2012}\rangle, & \mathbf{?m}, & \mathit{dm2}, & \langle \mathbf{?g},\ \tau\rangle &) \quad \wedge \\
\gamma(& \langle \mathbf{?o},\ \mathit{2012}\rangle, & \mathbf{?m}, & ?\mathbf{q}, & \langle \mathbf{?g},\ \tau\rangle &) \quad \wedge \\
\multicolumn{6}{l}{\mathbf{?o} \neq \mathit{Austria} \wedge \mathbf{?o} \uparrow \mathit{Austria} \wedge \mathbf{?q} \neq \mathit{dm2} \wedge \mathbf{?q} \sqsubseteq \mathit{dm2}}
\end{array}$$

This formula can be transformed to a database query. The results of this query are the possible instantiations of the template. A template is consistent if there is at least one instantiation. In our proof-of-concept prototype we implement this approach by generating a datalog program and evaluating it using the DLV system [6]. The datalog program implementing graph $\mathcal{G}$ can be found on the project website `http://www.dke.jku.at/research/projects/semcockpit.html`.

7 Related Work

Navigation modeling has received a lot of research interest in web engineering with the hypertext model of WebML [1] being the most prominent example. Sapia [3] introduces an approach for OLAP navigation modeling and query prediction. With regard to navigation modeling, BI Analysis Graphs go beyond Sapia's work by providing a more powerful conceptual query language (i.e., analysis situations) and a richer set of atomic navigation operations; further BI Analyis Graph Templates may be used in a more versatile manner since all parts of analysis situations may be defined as free variables. Romero et al. [4] introduce an approach to describe analytical sessions using a multidimensional algebra (MDA). Starting from a set of analytical SQL queries, they characterize and normalize these queries in terms of MDA and identify analytical sessions by computing similarities between related queries. Trujillo et al. [2] introduce an UML compliant approach for OLAP behavior modeling. In contrast to [4] and [2], our approach abstracts away from the organization of data in cubes and treats the calculation and aggregation of measure values as black box. Instead we talk of analysis situations which represent analytical queries in a declarative way. This allows us to focus our attention on the navigational behavior of business analysts and on the knowledge represented by this navigation, following our claim *'Navigation is Knowledge'*.

User-centric BI modeling has furthermore received research interest in terms of modeling preferences [7], personalization [8], query recommendations [9,10,11], and annotations [12]. Our work also draws ideas from active data warehouses, where analysis rules [13] were introduced to mimic the work of business analysts and from OLAP querying at the conceptual level as discussed by Pardillo et al. [14].

8 Summary and Future Work

In this paper we introduced BI Analysis Graphs and BI Analysis Graph Templates as the core of a conceptual formalism for sharing, understanding, and re-using navigational knowledge of business analysts. BI Analysis Graphs have evolved to a key ingredient of our project *Semantic Cockpit* [15]. In forthcoming phases of the project we will investigate extending BI Analysis Graphs with comparative scores (i.e., measures that quantify the relation between two independent points), performance and complexity of reasoning over BI Analysis Graph Templates, and using BI Analysis Graphs as one of the layers of a WebML-inspired conceptual modeling language for guided analytics applications.

References

1. Ceri, S., Brambilla, M., Fraternali, P.: The History of WebML Lessons Learned from 10 Years of Model-Driven Development of Web Applications. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 273–292. Springer, Heidelberg (2009)
2. Trujillo, J., Gómez, J., Palomar, M.S.: Modeling the Behavior of OLAP Applications Using an UML Compliant Approach. In: Yakhno, T. (ed.) ADVIS 2000. LNCS, vol. 1909, pp. 14–23. Springer, Heidelberg (2000)
3. Sapia, C.: On modeling and predicting query behavior in OLAP systems. In: DMDW, pp. 1–10 (1999)
4. Romero, O., Marcel, P., Abelló, A., Peralta, V., Bellatreche, L.: Describing Analytical Sessions Using a Multidimensional Algebra. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 224–239. Springer, Heidelberg (2011)
5. Lechtenbörger, J., Vossen, G.: Multidimensional normal forms for data warehouse design. Inf. Syst. 28(5), 415–434 (2003)
6. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM Trans. Comput. Log. 7(3), 499–562 (2006)
7. Golfarelli, M., Rizzi, S., Biondi, P.: myOLAP: An approach to express and evaluate OLAP preferences. IEEE Trans. Knowl. Data Eng. 23(7), 1050–1064 (2011)
8. Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D.: A personalization framework for OLAP queries. In: DOLAP, pp. 9–18 (2005)
9. Giacometti, A., Marcel, P., Negre, E., Soulet, A.: Query recommendations for OLAP discovery driven analysis. In: DOLAP, pp. 81–88 (2009)
10. Jerbi, H., Ravat, F., Teste, O., Zurfluh, G.: Preference-Based Recommendations for OLAP Analysis. In: Pedersen, T.B., Mohania, M.K., Tjoa, A.M. (eds.) DaWaK 2009. LNCS, vol. 5691, pp. 467–478. Springer, Heidelberg (2009)
11. Bentayeb, F., Favre, C.: RoK: Roll-Up with the K-Means Clustering Method for Recommending OLAP Queries. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 501–515. Springer, Heidelberg (2009)
12. Geerts, F., Kementsietsidis, A., Milano, D.: *i*MONDRIAN: A Visual Tool to Annotate and Query Scientific Databases. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 1168–1171. Springer, Heidelberg (2006)
13. Thalhammer, T., Schrefl, M., Mohania, M.K.: Active data warehouses: complementing OLAP with analysis rules. Data Knowl. Eng. 39(3), 241–269 (2001)
14. Pardillo, J., Mazón, J.N., Trujillo, J.: Extending OCL for OLAP querying on conceptual multidimensional models of data warehouses. Inf. Sci. 180(5), 584–601 (2010)
15. Neumayr, B., Schrefl, M., Linner, K.: Semantic Cockpit: An Ontology-Driven, Interactive Business Intelligence Tool for Comparative Data Analysis. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 55–64. Springer, Heidelberg (2011)

Ontology Alignment for Semantic Data Integration through Foundational Ontologies

Natalia Fonseca Padilha, Fernanda Baião, and Kate Revoredo

Federal University of the State of Rio de Janeiro (UNIRIO), Rio de Janeiro, Brazil
{natalia.padilha,fernanda.baiao,katerevoredo}@uniriotec.br

Abstract. Ontology alignment is the process of finding corresponding entities with the same intended meaning in different ontologies. In scenarios where an ontology conceptually describes the contents of a data repository, this provides valuable information for the purpose of semantic data integration which, in turn, is a fundamental issue for improving business intelligence. A basic, yet unsolved, issue in semantic data integration is how to ensure that only data related to the same real-world entity is merged. This requires that each concept is precisely defined into the ontology. From another perspective, foundational ontologies describe very general concepts independent of a particular domain, and precisely define concept meta-properties so as to make the semantics of each concept in the ontology explicit. In this paper, we discuss how the use of foundational ontology can improve the precision of the ontology alignment, and illustrate some examples using the UFO foundational ontology.

Keywords: Semantic integration, Foundational ontologies, Ontology alignment.

1 Introduction

Business data in enterprises is typically distributed throughout different but coexisting information systems. There is a manifold of applications that benefit from integrated information and the area of business intelligence (BI) is an example. In order to gain and maintain sustainable competitive advantages, integrated information can be used for OLAP querying and reporting on business activities, for statistical analysis and for the application of data mining techniques, towards improving decision-making processes.

In the enterprise context, the integration problem is commonly referred to as enterprise integration (EI), denoting the capability of integrating information and functionalities from a variety of information systems. Depending on the integration level, data and information or application logic level, EI is known respectively as Enterprise Information Integration (EII) or Enterprise Application Integration (EAI) [1].

In this paper, we focus on the integration of information and, in particular, the integration of data models, schemas, and data semantics, one of the several kinds of heterogeneity to be considered according to [1].

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 172–181, 2012.

Ontology alignment [9] is the process of finding related entities in two different ontologies. The most difficult integration problems are caused by semantic heterogeneity [1]. Semantic integration has to ensure that only data related to the same (or sufficiently similar) real-world entity is merged. In a context where each ontology conceptually describes the contents of its underlying data repository, techniques used for ontology alignment can be applied for data integration at the semantic level.

On the other hand, OntoUML is a conceptual modeling language designed to comply with the ontological distinctions and axiomatic theories put forth by the Unified Foundational Ontology (UFO) [8]. The OntoUML classes, for example, make the distinctions between an object and a process, types of things from their roles, among others, explicit.

In this paper, we show how the use of OntoUML improves the ontology alignment process for semantic data integration. A prerequisite for this is to address data semantic ambiguity by adding explicit metadata, which will be considered to identify and/or discard potential alignments.

This paper is structured as follows. Section 2 introduces the concepts of ontology and foundational ontologies. Section 3 describes how the techniques of ontology alignment face the problem of semantic integration. Section 4 introduces the conceptual modeling language OntoUML and some design patterns derived from the ontological foundations of this language and discusses how the use of OntoUML improves the alignment process. Section 5 reviews related works, followed by the conclusions in the Section 6.

2 Foundational Ontologies

An ontology is an explicit specification of a conceptualization [7]. Once represented as a concrete artifact, an ontology corresponds to the conceptual model of a domain of discourse, and supports communication, learning and analysis about relevant aspects of the underlying domain [8].

Four kinds of ontologies are distinguished according to their level of generality [6], as depicted in Figure 1.

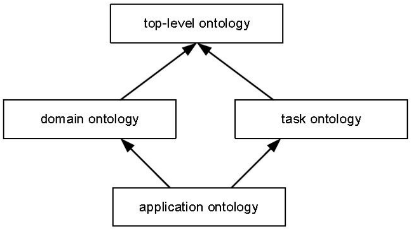


Fig. 1. Types of ontologies (Source: [6], p.7)

Top-level ontologies (also called upper-level ontologies or foundational ontologies) describe very general concepts which are independent of a particular problem or domain. Domain ontologies and task ontologies describe, respectively, the vocabulary related to a generic domain (independent of the activity being carried out) or a generic task or activity (independent of the domain on which it is applied). Application ontologies describe concepts depending both on a particular domain and task, which are often specializations of both the related ontologies. Foundational ontologies provide rigorous formal semantics for the high-level categories they describe and serve as a conceptual basis for domain ontologies [6].

For data integration purposes, the use of ontologies aims at ensuring semantic interoperability between data sources. This occurs because the semantics of data provided by data sources can be made explicit with respect to an ontology a particular user group commits to. Based on this shared understanding, the risk of semantic misinterpretations or false-agreements is reduced [13]. A typical strategy for semantic data integration is to apply ontology alignment techniques, as explained in the following section. The generality of the ontology alignment depends on the level of the ontology being considered.

3 Semantic Integration and Ontology Alignment

Ontology alignment is the process of finding corresponding entities (concept, relation, or instance) with related meaning (e.g. equality, subsumption) in two different ontologies. Aligning ontologies is a necessary condition to support semantic interoperability between systems, identifying relationships between individual elements of multiple ontologies [9].

The techniques used in the process of ontology alignment can be classified according to the granularity of the analysis (element-level or structure level) and according to the type of input (terminological, structural, extensional or semantic) [10].

Ontology alignment techniques that have an element-level analysis granularity and address semantic input may be based on foundational ontologies. These ontologies can be used as external sources of common knowledge and a key characteristic is that they are logic-based systems, and therefore, require semantic-based alignment techniques [10].

The development and improvement of techniques and tools for ontology alignment have been encouraged in recent years [11]. A variety of techniques are usually combined to calculate the degree of similarity between entities. However, in addition to the benefits for building conceptual models of a domain, foundational ontologies are still insufficiently explored in the ontology alignment literature.

Explicit and precise semantics of models are essential for reaching semantically correct and meaningful integration results. In data integration, the type of semantics considered is generally real-world semantics, which is concerned with the "mapping of objects in the model or computational world onto the real world, or the issues that involve human interpretation, or meaning and use of data or information" [3]. In [8], this requirement is called ontological adequacy, a measure of how close the models produced using a modeling language are to the situations in reality they are supposed to represent.

In the next section, we discuss how the use of OntoUML improves the ontological adequacy (and thus the alignment process) by resolving semantic ambiguity concerning data by explicit metadata.

By identifying the meta-categories from which the concepts are derived, it is possible to establish their nature, making the distinctions between an object and a process, types of things from their roles, among others, explicit. This distinction may help prevent incorrect associations in the alignment process, restricting the indication of equivalent terms to those derived from the same meta-category, i.e. those having the same conceptual nature [4].

4 Ontology Alignment through Foundational Ontologies

In this section we will discuss how the use of OntoUML improves the ontological adequacy and thus the alignment process by resolving semantic ambiguity concerning data by explicit metadata, which will be considered to identify and/or discard potential alignments.

4.1 OntoUML

OntoUML is a conceptual modeling language designed to comply with the ontological distinctions and axiomatic theories put forth by the Unified Foundational Ontology (UFO) that results from a re-design process of the Unified Modeling Language (UML).

UFO is a foundational ontology that has been developed based on a number of theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology [8]. It is composed by three main parts. UFO-A is an ontology of endurants (objects). UFO-B is an ontology of perdurants (events, processes). UFO-C is an ontology of social entities (both endurants and perdurants) built on the top of UFO-A and UFO-B.

A fundamental distinction in UFO is between particulars and universals. Particulars are entities that exist in reality possessing a unique identity, while universals are patterns of features, which can be realized in a number of different particulars. Class diagrams are intended to represent the static structure of a domain and should always be interpreted as representing endurant universals. A UML profile proposed in [8] is a finer-grained distinction between different types of classes to represent each of the leaf ontological categories specializing substantial universal types of UFO-A, as depicted in Figure 2.

Substantials are entities that persist in time while keeping their identity (as opposed to events such as a business process or a birthday party). Constructs that represent *Sortal Universals* can provide a principle of identity and individuation for its instances. *Mixin Universal* is an abstract metaclass that represents the general properties of all mixins, i.e., non-sortals (or dispersive universals). *Phase* represents a sortal instantiated in a specific world or time period, but not necessarily in all of them (such as a child, adolescent and adult phases of a Person). *Role* represents a sortal that may or may not be instantiated but, once it is, this depends on its participation in an event or in a specific relation. A role defines something which may be assumed in a world, but not necessarily in all possible worlds (such as a student or professor role played by a Person). Due to space restrictions, we will not define all other OntoUML categories.

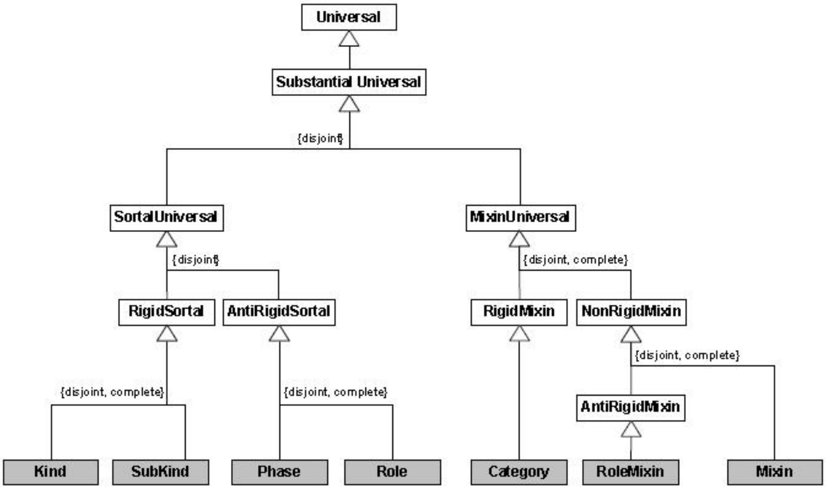


Fig. 2. Ontological Distinctions in a Typology of Substantial Universals (Source: [8], p. 106)

In the next section we will detail some meta-properties of the Phase and Role constructs and present some design patterns derived from these meta-properties. In section 4.3 we illustrate how OntoUML improves the ontology alignment considering these two constructs.

4.2 Design Patterns

In this section we introduce two design patterns derived from the ontological foundations of OntoUML: The Role Design Pattern and The Phase Design Pattern [2, 8].

We start by making a basic distinction between categories considering the Rigidity meta-property. A rigid universal is one that applies to its instances necessarily, i.e., in every possible world. A type T is rigid iff for every instance x of that type, x is necessarily an instance of that type. In contrast, a type T' is anti-rigid iff for every instance y of T', there is always a possible world in which y is not an instance of T'. An example of this distinction can be found by contrasting the rigid type Person with the anti-rigid type Customer. A postulate derived from this meta-property says that a class representing a rigid universal cannot specialize (restrict) a class representing an anti-rigid one. Additionally, every individual in a conceptual model of the domain must be an instance of one (and only one) rigid sortal.

Roles and phases are anti-rigid sortal types. The instances can move in and out of the extension of these types without any effect on their identity. However, while in the case of phase these changes occur due to a change in the intrinsic properties of these instances, in the cases of role they occur due to a change in their relational properties. A constraint of the anti-rigid sortal says that for every role or phase x there is a unique kind k such that k is a supertype of x.

Role Design Patterns. A Role possesses a meta-property (absent in Phase) named Relational Dependence. As a consequence, a OntoUML class stereotyped as «role» must always have as supertype a kind and be connected to an association representing this relational dependence condition, as depicted in Figure 3.

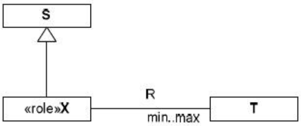


Fig. 3. The Role Design Pattern (Source: [2], p. 4)

A recurrent problematic case in the literature of role modeling is termed the problem of role with multiple disjoint allowed types [8]. In the model of Figure 4(a), the role Customer is defined as a supertype of Person and Organization. This modeling choice violates the postulate derived from the Rigidity meta-property discussed above and produces an ontologically incorrect conceptual model.

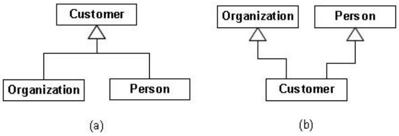


Fig. 4. Problems with modeling roles with multiple allowed types (Source: [8], p. 281)

Firstly, not all persons are customers, i.e., it is not the case that the extension of Person is necessarily included in the extension of Customer. Moreover, an instance of Person is not necessarily a Customer. Both arguments are equally valid for Organization. In the model of Figure 4(b), instead, the Customer is defined as a subtype of Organization and Person. However, more than one rigid sortal (in this example represented by the classes Organization and Person) could not apply to the same individual, what leads to a design pattern depicted in Figure 5.

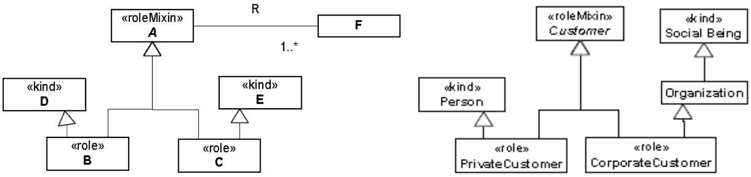


Fig. 5. The Role Modeling with Disjoint Admissible Types Design Pattern (Source: [8], p. 282)

In Figure 5, the application of the design pattern is illustrated by an instantiated model of the domain discussed above. The abstract class A is the role mixin that covers different role types (e.g. Customer). Classes B and C are the disjoint subclasses of A, representing the sortal roles that carry the principles of identity that govern the individuals that fall in their extension (e.g. PrivateCustomer and CorporateCustomer). Classes D and E are the kinds that supply the principles of identity carried by B and C, respectively (e.g. Person and Organization). The association R and the class F represents the relational dependence condition (which is not represented in the instantiated model but could be a purchase association with an enterprise).

Phase Design Pattern. A phase represents the phased-sortals phase, i.e. anti-rigid and relationally independent universals defined as part of a partition of a sortal. As depicted in Figure 6, the parts are disjoint (mutually exclusive) and complete. One example is the kind Person, restricted by a phase-partition ⟨Child, Adolescent, Adult⟩

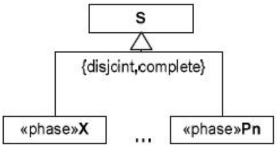


Fig. 6. The Phase Design Pattern (Source: [2], p. 4)

4.3 OntoUML Improving the Ontology Alignment

The Role Design Pattern induces the modeler to make explicit design features that would be implicit in a UML model. By stereotyping a class as a <<role>>, the modeler brings additional information intrinsic to this construct and is oriented to make both its kind supertype and its relational dependence condition explicit in the model.

Therefore, to ensure that the alignment between roles in two different ontologies is semantically correct, it is necessary to observe if the principle of identity (explicited by their kind superclasses) and their relational dependency are similar.

Consider an enterprise that has two subsidiaries, which operate independently from one another. In one subsidiary, customers may be private or corporate and in the otheronly private. A simplified model of this reality is represented in Figure 7, where dotted lines are possible concept alignments that are further investigated as follows.

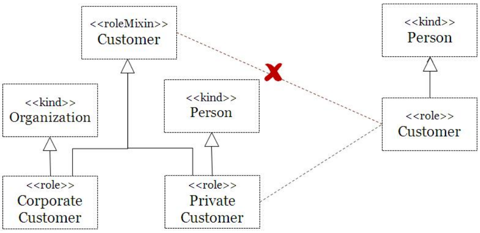


Fig. 7. Investigating the alignment of <<role>> classes

In this example, although there is a Customer concept in each ontology (which would probably be aligned using manual analysis or lexical-based techniques), the <<roleMixin>> stereotype applied to the Customer concept in the left ontology makes it explicit that it is not semantically equivalent (and therefore should not be aligned) to the class Customer in the right ontology, which is stereotyped as a <<role>>. Consider a decision-making scenario in which a manager requests an employee to gather data comparing the sales performance of both subsidiaries of the organization, so as to decide where to allocate marketing investments focusing on private customers. By taking foundational semantics and the role design pattern into account, the employee concludes that both Customer classes cannot be aligned to each other, and proceeds by searching for the <<kind>> concept that is the most specific superclass of the <<role>> concept in the left ontology (that is, Person). The employee then identifies the <<role>> subclass PrivateCustomer and considers it as being equivalent to the <<role>> Customer in the right ontology (even though their names do not match). Considering the Role Design Patterns presented in the previous section, <<role>> and <<roleMixin>> classes should be connected to an association representing their relational dependence condition. In this example, this is not explicit to simplify the analysis. However, this is an additional aspect to be observed when analyzing the alignment alternatives.

The Phase Design Pattern induces the modeler to make all parts of a phase partition explicit. Phases are relationally independent universals; moreover, any change in and out of the extension of these types occurs due to a change in the intrinsic properties of these instances. Because of these meta-properties, we state that if two kinds are aligned and both have phases that refer to the same intrinsic property, then the phases should be aligned with each other. Similarly, if we have a case in which the phases refer to the same intrinsic property but the alignment between them is incomplete, then the alignment between the kinds is incorrect.

In the example of Figure 8, the right ontology explicit that an embryo is already considered a Person while in the left ontology a person is considered a creature that can be in a phase of embryo or person. This semantic difference must be considered in the alignment process.

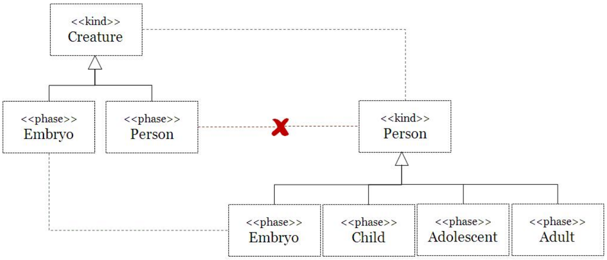


Fig. 8. Investigating the alignment of <<phase>> classes

Although there is a Person concept in each ontology, the stereotype <<phase>> applied to the Person concept in the left ontology makes it explicit that it is not semantically equivalent to the class Person in the right ontology, which is stereotyped as a

<<kind>>. By taking foundational semantics and the phase design pattern into account, the next step is to search for the <<kind>> concept that is the most specific superclass of the <<phase>> concept in the left ontology (that is, Creature). If the Child, Adolescent and Adult phases in the right ontology are comprehended in the Person phase of the left ontology, then it is possible to further recommend the alignments: Creature to Person, Embryo to Embryo and Person to (Child + Adolescent + Adult).

5 Related Work

One main point that has guided the development of the approach presented in [4] is the use of foundational ontologies. To establish the relationship among foundational ontologies and domain ontologies, for each first-level concept at the domain ontology, a foundational concept was associated. Thus, the result is a unique integrated ontology, composed by the domain ontology and some of the meta-categories of a foundational ontology. This information was relevant for the taxonomic similarity measure, as it becomes possible to compare upper-level concepts in the hierarchy when a candidate pair of concepts is under analysis.

The approach presented in this paper, in turn, proposes the direct use of meta-properties of the constructs, with the determination of indicative and restrictive rules to be applied during concept alignment.

Other works address foundational ontologies in the context of ontology alignment but they are more directly related to the use of reference ontologies to support the alignment of other ontologies on the same domain. In [12] the techniques applied to associate the classes of the domain ontologies to the classes of the foundational ontologies are typically used to associate concepts of domain ontologies. A higher precision was obtained with foundational ontologies that include many domain-specific concepts in addition to the upper-level ones. In [5] the hypothesis is that a domain reference ontology that considers the ontological distinctions of OntoUML can be employed to achieve semantic integration between data standards. The hypothesis is tested by means of an experiment that uses an electrocardiogram (ECG) ontology and conceptual models of the ECG standards. The authors advocate that such a principled ontological approach is of value for the semantic integration reported in the work.

6 Conclusion and Future Work

Information integration is still a challenge, especially when considering semantic issues. The process of ontology alignment is a condition to support semantic interoperability between systems, identifying relationships between individual elements of multiple ontologies. Explicit and precise semantics of models are essential for semantically correct and meaningful integration results and thus a main issue for the decision making process based on integrated data.

In this paper we discussed how the use of OntoUML improves the ontological adequacy of an ontology and thus the alignment process by resolving semantic ambiguity concerning data by explicit metadata, considered to identify and/or discard potential alignments, improving the precision of the result.

The examples discussed above demonstrated how the stereotype of the OntoUML classes indicated incorrect associations in the alignment process (which would probably be aligned using manual analysis or lexical-based techniques) and how the meta-properties of the constructs lead to the correct alignments.

Future work includes formalization of indicative and restrictive rules based on the meta-properties of a larger set of constructs of OntoUML to be applied during the alignment process. Moreover, the automatization of the proposal and its application in real scenarios will also be considered.

References

1. Ziegler, P., Dittrich, K.R.: Data Integration - Problems, Approaches, and Perspectives. In: Krogstie, J., Opdahl, A.L., Brinkkemper, S. (eds.) Conceptual Modelling in Information Systems Engineering, pp. 39–58. Springer, Heidelberg (2007)
2. Guizzardi, G., Graças, A.P., Guizzardi, R.S.S.: Design Patterns and Inductive Modelling Rules to Support the Construction of Ontologically Well-Founded Conceptual Models in OntoUML. In: 3rd International Workshop on Ontology-Driven Information Systems (ODISE 2011), London, UK (2011)
3. Ouksel, A.M., Sheth, A.P.: Semantic Interoperability in Global Information Systems: A Brief Introduction to the Research Area and the Special Section. SIGMOD Record 28(1), 5–12 (1999)
4. Silva, V.S., Campos, M.L.M., Silva, J.C.P., Cavalcanti, M.C.: An Approach for the Alignment of Biomedical Ontologies based on Foundational Ontologies. Journal of Information and Data Management 2(3), 557–572 (2011)
5. Gonçalves, B., Guizzardi, G., Pereira Filho, J.G.: Using an ECG reference ontology for semantic interoperability of ECG data. Journal of Biomedical Informatics 44, 126–136 (2011)
6. Guarino, N.: Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction and Integration (1998)
7. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies 43(5/6), 907–928 (1995)
8. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models Ph.D. Thesis, University of Twente, The Netherlands (2005)
9. Ehrig, M.: Ontology Alignment: Bridging the Semantic Gap. Springer (2007)
10. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer (2007)
11. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology Alignment Evaluation Initiative: Six Years of Experience. In: Spaccapietra, S. (ed.) Journal on Data Semantics XV. LNCS, vol. 6720, pp. 158–192. Springer, Heidelberg (2011)
12. Mascardi, V., Locoro, A., Rosso, P.: Automatic Ontology Matching Via Upper Ontologies: A Systematic Evaluation. IEEE Transactions on Knowledge and Data Engineering 22(5), 609–623 (2010)
13. Ziegler, P., Dittrich, K.R.: User-Specific Semantic Integration of Heterogeneous Data: The SIRUP Approach. In: Bouzeghoub, M., Goble, C.A., Kashyap, V., Spaccapietra, S. (eds.) ICSNW 2004. LNCS, vol. 3226, pp. 44–64. Springer, Heidelberg (2004)

Preface to the Fourth International Workshop on Requirements, Intentions, and Goals in Conceptual Modeling (RIGiM 2012)

The use of intentional concepts, the notion of "goal" in particular, has been prominent in recent approaches to requirements engineering, producing a body of work focusing on Goal-Oriented Requirements Engineering (GORE). RIGiM (Requirements, Intentions, and Goals in Conceptual Modeling) aims to provide a forum for discussing the interplay between requirements engineering and conceptual modeling, and in particular, to investigate how goal- and intention-driven approaches help in conceptualising purposeful systems. What are the upcoming modelling challenges and issues in GORE? What are the unresolved open questions? What lessons are there to be learnt from industrial experiences? What empirical data are there to support the cost-benefit analysis when adopting GORE methods? Are there applications domains or types of project settings for which goals and intentional approaches are particularly suitable or not suitable? What degree of formalization, automation or interactivity is feasible and appropriate for what types of participants during requirements engineering?

This year, RIGIM includes a keynote on requirements for adaptive systems. All papers were reviewed and evaluated by three reviewers from the program committee. Papers with strongly conflicting reviews received a round of online discussion. Of the eight papers submitted, three were accepted for inclusion in the proceedings and for presentation at the workshop, resulting in an acceptance rate of 38%.

Workshop Programme

Keynote: Requirements in the Land of Adaptive Systems

Speaker: John Mylopoulos, University of Trento

Abstract

Adaptive systems of any sort (software, hardware, biological or social) consist of a base system that carries out activities to fulfill some requirements R, and a feedback loop that monitors the performance of the system relative to R and takes corrective action if necessary. We adopt this view of adaptivity for software-intensive systems and sketch a framework for designing adaptive systems which starts with requirements models, extends them to introduce control-theoretic concepts, and uses them at run-time to control the behaviour of the base system. We also present preliminary results on the design of adaptive systems-of-systems where the main problem is how to maintain alignment between a collection of independently evolving systems so that they continue to fulfill a set of global requirements.

The presentation is based on joint research with Vitor Souza, Alexei Lapouchnian, Fatma Aydemir and Paolo Giorgini all with the University of Trento.

Speaker Biography

John Mylopoulos holds a distinguished professor position (chiara fama) at the University of Trento, and a professor emeritus position at the University of Toronto. He earned a PhD degree from Princeton University in 1970 and joined the Department of Computer Science at the University of Toronto that year. His research interests include conceptual modelling, requirements engineering, data semantics and knowledge management. Mylopoulos is a fellow of the Association for the Advancement of Artificial Intelligence (AAAI) and the Royal Society of Canada (Academy of Sciences). He has served as programme/general chair of international conferences in Artificial Intelligence, Databases and Software Engineering, including IJCAI (1991), Requirements Engineering (1997), and VLDB (2004). Mylopoulos was recently awarded an advanced grant from the European Research Council for a project titled "Lucretius: Foundations for Software Evolution".

Paper Session:

- Itzel Morales-Ramirez, Matthieu Vergne, Mirko Morandini, Luca Sabatucci, Anna Perini and Angelo Susi. *Where did the requirements come from? A retrospective case study*
- Alberto Siena, Silvia Ingolfo, Angelo Susi, Ivan Jureta, Anna Perini and John Mylopoulos. Requirements, Intentions, *Goals and Applicable Norms*
- Daniel Amyot, Azalia Shamsaei, Jason Kealey, Etienne Tremblay, Andrew Miga, Gunter Mussbacher, Mohammad Alhaj, Rasha Tawhid, Edna Braun and Nick Cartwright. *Towards Advanced Goal Model Analysis with jUCM-Nav*

We would like to thank all authors, presenters, and reviewers for their continued work in forming the conceptual modelling foundations for goal-oriented techniques. We are also grateful to the ER 2012 workshop chairs for giving us the opportunity for the continuation of the RIGiM workshops.

October 2012

Colette Rolland
Jennifer Horkoff
Eric Yu
Camille Salinesi
Jaelson Castro

Where Did the Requirements Come from? A Retrospective Case Study

Itzel Morales-Ramirez, Matthieu Vergne, Mirko Morandini,
Luca Sabatucci, Anna Perini, and Angelo Susi

Center for Information and Communication Technology, FBK-ICT
Via Sommarive, 18, 38123 Trento (I)
{imramirez,vergne,morandini,sabatucci,perini,susi}@fbk.eu

Abstract. Understanding complex organisations in terms of their stakeholders' goals, intentions and resources, is a necessary condition for the design of present day socio-technical systems. Goal-oriented approaches in requirements engineering provide concepts and techniques to support this analysis. A variety of goal-oriented modelling methods are available, together with guidelines for their application, as well as real case studies success stories.

Our long term research objective is to derive useful suggestions for practitioners about which information sources are more promising for performing effective goal-oriented analysis and requirements elicitation of a complex domain, as well as about possible limits and pitfalls. As a first step towards this objective we perform a retrospective case study analysis of a project in the domain of ambient assisted-living residences for people affected by Alzheimer's.

In this paper we describe the design of this study, present an analysis of the collected data, and discuss them against the proposed research questions, towards investigating the effectiveness of information sources for goal modelling and requirements elicitation in complex domains.

Keywords: Requirements Engineering, Requirements Elicitation Techniques, Goal-Oriented Modelling.

1 Introduction

Software systems for complex organisations are conceived as socio-technical systems (STSs), systems in which human and technological aspects are strongly interrelated. Eliciting the requirements for such systems builds upon a deep understanding of the involved human organisations in terms of the stakeholders' goals, intentions and resources, and of the role of technology towards enabling the achievement and maintenance of those goals.

Goal-oriented (GO) approaches in requirements engineering provide concepts and techniques to model social dependencies and to perform goal analysis, thus adopting a GO approach seems to be a natural choice. Experiences collected in complex real projects give evidence that different elicitation techniques need to be combined in order to better exploit the different sources of domain information and to model the various types of knowledge that characterise an STS domain.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 185–194, 2012.

The problem we face when starting a new project for developing an STS is how to identify useful domain knowledge sources and how to select the appropriate techniques for capturing knowledge and building an effective GO model for the intended STS. This relates to the requirements elicitation problem, which is largely addressed by the Requirements Engineering research community [9,4,8,6].

The long term objective of our research is to derive useful suggestions for practitioners about which information sources, among stakeholder interviews, domain documents, observations, etc., are more promising for performing an effective GO analysis of a complex domain, as well as about possible limits and pitfalls.

As a first step towards this objective we revisit our experience in applying GO approaches in real projects. Specifically, we investigate whether it is possible to derive empirical data about which information sources supported activities of modelling actors, goals, tasks, resources, and strategic dependencies and which knowledge elicitation strategy guided domain analysis and requirements collection, performing a retrospective analysis [10] of the *ACube* (Ambient Aware Assistance) project[1]. The *ACube* project application domain concerns an assisted-living residence called *Social Residence* for elderly people suffering Alzheimer's disease, who need continuous but unobtrusive monitoring of a variety of health-related issues. Worth mentioning is the heterogeneity of the stakeholders of social residences, including patients and their relatives, social workers, managers of the sanitary structure and nurses.

In our study, we perform a retrospective analysis of the project documentation, including the elicitation techniques, the trace links between requirements and goals, and the elicited set of requirements. This analysis is guided by three research questions. Moreover, two authors of this paper, who were acting as project analysts, were available for clarifying findings to the authors performing this project review.

The paper is organised as follows. In Section 2, we give an overview of the *ACube* project and of the requirements elicitation process that was adopted. In Section 3 we sketch the design of the proposed empirical analysis and the possible measures, to investigate on the use of the different information sources in GO modelling of a complex domain. First findings, extracted from the available documentation, are presented and discussed in Section 4. Related work is presented in Section 5, while Section 6 draws the conclusions and points out future work directions.

2 The *ACube* project

The *ACube* project aimed at developing an advanced, generic monitoring infrastructure for Assisted-Living, able to monitor in a uniform, adaptive, and high quality manner the patients of a social residence, the environment and its operators, and the ongoing activities, thus realising a highly developed smart environment as a support to medical and assistance staff.

The solution developed in the project exploits low energy consumption wireless networks of sensors and actuators. The resulting system, sketched in Figure 1, is based on: a set of sensors and actuators, which are distributed in the environment — e.g.

[1] The project was funded by the Autonomous Province of Trento in Italy (2008-2011). Detailed information about the *ACube* project can be found at http://acube.fbk.eu/.

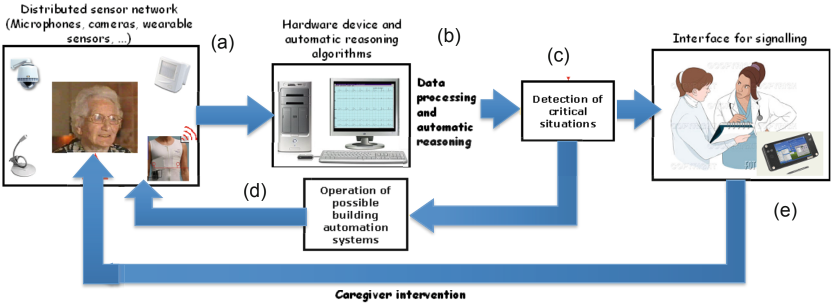

Distributed sensor network (Microphones, cameras, wearable sensors, ...)
(a)
Hardware device and automatic reasoning algorithms
(b)
Data processing and automatic reasoning
(c)
Detection of critical situations
Interface for signalling
(d)
Operation of possible building automation systems
(e)
Caregiver intervention

Fig. 1. A vision of the *ACube* system

microphones, cameras and alarms — or embedded in patients' clothes — e.g. biological sensors for ECG (see label (a) in Figure 1), and algorithms devoted to the higher level functions to assess monitored data and discover critical situations (see label (b) in Figure 1), which trigger configured actuators (d) or alarms calling for human operator intervention (e). All events are recorded for later debriefing by human operators. The communication infrastructure is designed for a high degree of configurability allowing to add new sensors to the system or to dynamically switch on and off sensors and actuators to save energy. This technology should allow an unobtrusive monitoring of the social residence guests.

2.1 Requirements Elicitation Artefacts and Process

In *ACube* an activity of paramount importance was the analysis of the requirements of the system, with the need of managing the trade-off between cost containment and improvement of quality of services in a specialised centre for people with severe motor or cognitive impairments, such as a social residence for elderly people. The project consortium had a multidisciplinary nature, involving software engineers, sociologists and analysts. Moreover, social residence professionals representing end users were directly engaged in design activities.

The joint use of both approaches User Centred Design [3] and Goal Oriented Requirements Engineering [5] allowed us to manage the multidisciplinary knowledge between stakeholders by balancing their needs and technical constraints, and in parallel by ensuring the validity, completeness and traceability of requirements. The requirements analysis phase of the project had a strict deadline of six month due to the schedule of the project, after which the technological team received the requirements in order to start the development.

The major sources of information in the project were the interviews with the domain stakeholders (in particular operators, doctors and managers), brainstorming sessions and domain document, such as the *Carta dei Servizi*, which describes the services the social residence is committed to give to the patients and to their families (such as reports on the condition of the patient) and the major activities to be performed to set up these services.

The major results of the elicitation and analysis phase were the definition of four different macro-services that the *ACube* system might provide: (i) "localisation and

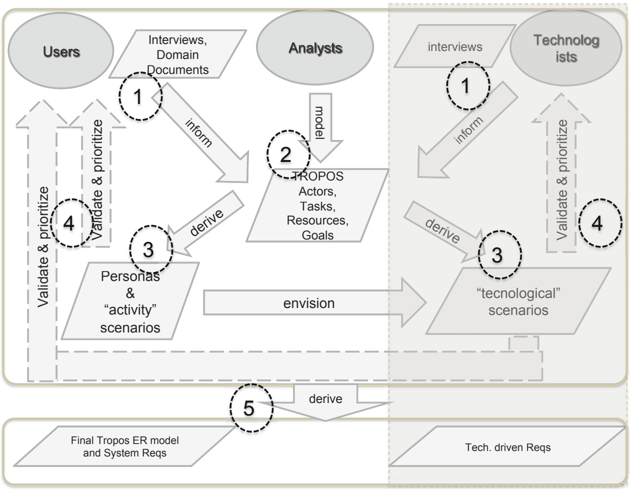


Fig. 2. A sketch of the *ACube* requirements elicitation and validation process

tracking of the patients and operators in the residence", (ii) "identification of the behaviour of the patients", (iii) "coordination of caregivers activity with a (semi) automatic report system" and (iv) "therapy management and administering".

Out of these scenarios and of the Tropos requirements diagrams a set of functional and non-functional requirements was generated. A first validation session was held with 27 researchers. A second validation session was organised with some of the stakeholders, including 3 managers and 8 operators of nursing homes previously involved in the early exploration phase. The goal of these sessions was the assessment of the validity, acceptability and feasibility of the requirements.

Most of the techniques and information sources used during the project, for eliciting, collecting and modelling data, belong to user centred design approach as well as goal oriented technique. In particular, we performed *an analysis of the existing documentation*, conducted *interviews* with domain stakeholders, led *brainstorming* to have feedback on the analysis of the domain and on the envisaged solutions, and modelled the domain via *goal-oriented requirements engineering* technique, by adopting the Tropos methodology [5].

The process followed in *ACube*, sketched in Figure 2, involves three roles — Users, Analysts, and Technologists — and can be divided into five main phases [7].

Analysis of the domain. Here a first activity of *analysis of the existing documentation* was performed, in particular of the domain document (see label "1" in Figure 2). Moreover, *unstructured and structured interviews* (also via questionnaires) with managers, doctors and caregivers were performed. In particular three representative sites (of different sizes) were selected for the research, resulting in 4 interviews with managers and 8 interviews with caregivers. The objective was to gather data directly from the context, to keep the richness of the data and avoid abstraction at the requirements level of the analysis, and to make analysts and stakeholders collaborate in understanding the domain.

Data interpretation and modelling. The data interpretation and modelling performed by the analysts, via goal oriented techniques, is the step in which data coming from the domain is shared across the team and becomes knowledge (label "2" in Figure 2). In our process, data interpretation was concurrently carried out in a twofold way: (i) domain context analysis, and (ii) early requirements phase of Tropos to model retrieved information and to state hypotheses about the existence of entities (mainly goals and actors). Here the Tropos early requirements phase was executed in four iterations characterised by an increasing precision of the model and the reduction of open points that were clarified by using other techniques. The previous versions of the Tropos model were a source of information for the analysts to refine the subsequent versions.

Specification of user (activity) and technological scenarios. To obtain feedback from users and technologists, user scenarios were also used to envisage the technological scenarios (label "3").

Feedback via brainstorming sessions from both stakeholders of the domain and researchers. This activity allowed to confirm the validity of the retrieved models via feedback, from the domain stakeholders and researchers, and new iterations of contextual inquiry and questionnaires (label "4").

Retrieve system requirements and technical requirements. The model and list of requirements were released together with the final version of the Early and Late Requirements Tropos model (label "5" in Figure 2).

In the following we focus on the part of the process on the left in Figure 2, involving the interaction of users and analysts for the specification of system requirements.

3 Empirical Study Design

We perform a retrospective analysis of the *ACube* project by evaluating the available documentation along the following three questions:

- RQ1. Which information sources, among stakeholder interviews and domain documents, are relevant for the different types of knowledge captured in early-requirements goal models?
- RQ2. How did the different information sources contribute to model elements in different abstraction levels of a GO model?
- RQ3. In which way did goal models and information sources contribute to the elicitation of system requirements?

Measures. To investigate the first research question we use a quantitative analysis of a set of project's requirements artefacts, complemented by clarifications on specific aspects, which are asked directly to two project analysts. The quantitative analysis is carried out on the output of the *ACube* early requirements model delivered as tables with lists of entities, which were validated by domain stakeholders, and on the trace links from goal model elements to information sources, which were recorded during requirements analysis. We try to understand the major information sources for the elicitation of these elements, among the eight recorded interviews with domain stakeholders (2 managers, 1 nurse, 3 social workers, and 2 specialised collaborators), the available domain document, here the *Carta dei Servizi* of the social residence, and a preliminary version of an early requirements model. Moreover, the goal analysis itself, performed iteratively by organising goals and putting them in context, is an important source for new goal model elements. By analysing the trace links between goal model elements and the information sources, we count how many goals can be traced back to one or more among the above 10 information sources. We repeat this counting for actors, tasks, and resources. When no trace links are found, the original analysts are asked for clarifications.

The second research question is approached by trying to rebuild the early requirements goal model, with its hierarchies and dependencies, from the available goals and actors lists. Associating this goal model with the information sources and analysing the positions of the goals which emerged during the iterative construction of the goal model (source *Tropos ER model* in Table 1), detailed conclusions can be drawn on the goal-oriented elicitation process.

To analyse the third question we consider the early-requirements goal model, which has been validated by domain stakeholders, and the list of 78 requirements (of which 57 are functional) as the output of the *ACube* requirements elicitation process[2] illustrated in Figure 2. For each requirement we check the recorded links to goals in the validated early requirements model and transitively obtain the underlying information source. An analysis of the distribution of sources, actors, goals and plans is then made, to draw conclusions on the elicitation process. If there are no recorded links, we consider the following cases: i) check if the requirement refers to a task in the model, whose trace link was not recorded because an explicit means-end relationship between the task and a goal in the model was missing, or ii) enquire the analysts about possible mistakes.

4 Data and Analysis

We follow the analysis of the available *ACube* documents as described in the Section 3 and document the results.

The number of Tropos actors, tasks, resources and goals retrieved from the various information sources: from domain document (*Carta dei Servizi*), from interviews, and during the Tropos Early Requirements analysis, are reported in Table 1. The total number of entities, and the number of model entities that have more than one source, are also recorded. In general, in the analysed social residence domain, interviews produced the major part of elements in the early-requirements goal model.

Looking deeper at the results, we notice however that, for the activities to be performed, the domain document was the major source of information. This finding can be explained considering the fact that the activities represent services that are offered by the actors of the residence to the patients and to the external actors (such as families and control authorities), which are prescribed at the organisational and governmental level and that are mainly reported in the *Carta dei Servizi*. The remaining activities, extracted via interviews, are mainly internal and are necessary to provide the services described in the documents.

[2] Notice that for this study we are not considering the technological requirements, which are also part of the output of *ACube*.

Table 1. Contribution of information sources for modelling the Tropos elements

Information source Goal model elements	actors	activities	resources	goals	sum
Domain Document *Carta dei Servizi*	5	24	3	3	35
Interviews	18	15	18	10	61
Tropos Early Requirements Model	0	0	0	12	12
Total number of elements used in the Tropos model	20	27	19	24	90
Elements found using more than one source	3	12	2	1	18

Regarding the actors, only few of them are extracted from the domain document, since a social residence has the freedom to establish by itself several roles in the organisation, and only few roles are fixed at governmental or institutional level. Looking at the single interviews, most actors are added in the first two (held with the coordinators of the structure), which seems reasonable, since these stakeholders know the organisational structure at best. In contrast, most interview partners mentioned resources needed for their work, thus they were added to the domain model quite uniform throughout the interviews.

Concerning the goals, they were retrieved from various sources, in particular from the interviews with the coordinators. However, also a specialised worker, the physiotherapist, gave rise to nearly 15% of the goals, while the social workers did not directly help to reveal new goals. Twelve goals were retrieved indirectly, during the goal analysis phase (i.e. in the Tropos Early Requirements Model). With the following analysis we are able to specify their source more precisely.

For answering to the second research question, we rebuilt the early requirements goal model, collecting the textual information available and the recorded goal dependencies, and annotating the artefacts with their original information source. Both the high-level goals and the leaf tasks (*activities*) are discovered already through interviews and domain document. Out of the 12 goals which emerged only during the analysis, 7 were internal goals added to create links between tasks and high-level goals, and the 5 remaining goals were introduced bottom-up, as motivation for an activity. The mix of top-down and bottom-up elicitation confirms the method proposed by Giorgini et al. [5], in contrast to previous guidelines.

From this analysis we can also state that the various layers of the goal model have been built exploiting the sources of information as reported in Figure 3: while the top and bottom layers of the model have their source mainly in domain document and interviews, the internal parts are often tacit knowledge [8], which seems either too "obvious" or too "abstract" to the stakeholders, and has thus often to be discovered by the analyst during goal modelling.

For answering to the third question we analysed the requirements document provided as output of the *ACube* project, which defines specific goals as the *motivation* for requirements. Joining information sources, requirements and the goal models obtained in the precedent analysis, we obtain an overview over the sources involved in the elicitation of requirements and the distribution of the various artefacts, which leads to various observations. Table 2 shows that 40% of the requirements were motivated by only 2 of the 28 goals. Also, all the requirements are associated to goals of only two actors, the *responsible* and the *social operator*. This can be explained by the specific aim of the project, which was devoted to support the social operators in their daily work. We

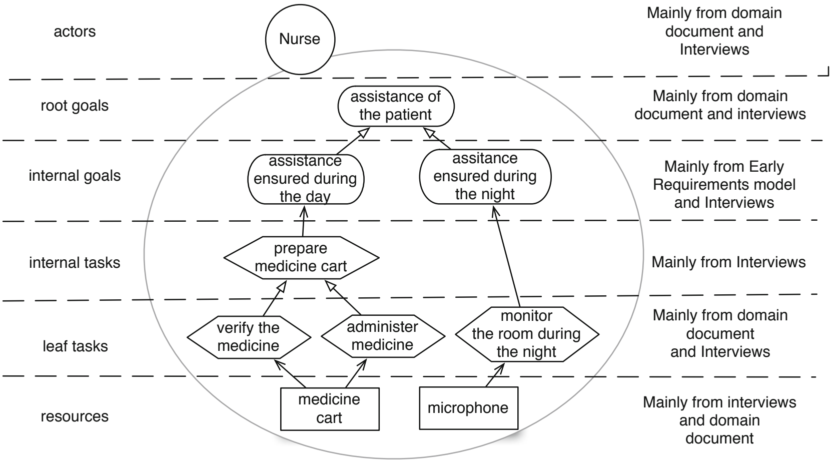

actors
Nurse
Mainly from domain document and Interviews
root goals
assistance of the patient
Mainly from domain document and interviews
internal goals
assistance ensured during the day
assitance ensured during the night
Mainly from Early Requirements model and Interviews
internal tasks
prepare medicine cart
Mainly from Interviews
leaf tasks
verify the medicine
administer medicine
monitor the room during the night
Mainly from domain document and Interviews
resources
medicine cart
microphone
Mainly from interviews and domain document

Fig. 3. Excerpt of a Tropos diagram representing a nursing home, with an explanation of the various goal model elements and the associated major sources of information in *ACube*.

omitted the non-functional requirements, since, for most of them, no motivating goals were defined.

Looking at the transitive relationship between sources, goals and requirements reveals that most requirements (except the ones motivated by goals elicited during the analysis) arose from the interviews with the responsible and the physiotherapist.

However, these findings have to be critically examined: the goals attributed to some interview were often very general, such as "act promptly in critical situations". In a second step, they can lead to various requirements which have few in common with the situation described in the original interview. This effect of goal modelling can be observed mainly due to the very condensed description of goals in a goal model and the missing (graphical) link to the information sources. Thus, these goals will be perceived by the analyst from a more abstract, high level viewpoint, and decomposed and

Table 2. Goals with the relative actors and sources, together with the number of requirements in which they are cited as motivation (only goals with a number of requirements ≥ 1 are shown).

Goal	Actor	Source	# of functional req.
G01 (provide nursing care)	A10 (social operator)	Interv. to coordinator	5
G07 (guarantee safety)	A10 (social operator)	Interv. to coordinator	13
G09 (optimise resources)	A03 (responsible)	Interv. to responsible	1
G10 (intervene promptly)	A10 (social operator)	Interv. to physiother.	11
G14 (improve the quality of service)	A03 (responsible)	Early Req. analysis	2
G15 (guarantee continuity of the service)	A03 (responsible)	Early Req. analysis	1
G16 (promote teamwork)	A03 (responsible)	Early Req. analysis	7
G17 (promote service personalisation)	A03 (responsible)	Early Req. analysis	6
G21 (manage emergency situations)	A07 (medical doctor)	Early Req. analysis	2
G22 (provide clinical surgery)	A07 (medical doctor)	Early Req. analysis	4
G23 (guarantee continuity of clinical surgery)	A15 (relatives)	*Carta dei Servizi*	2
G27 (manage clinical emergency)	A04 (guest)	Early Req. analysis	3
		Total	57

operationalised accordingly. Moreover, the reliability of the available trace links was not verified and could thus be a serious threat to validity for the whole analysis.

Three out of the 78 identified requirements did not have any direct link to goals or information sources. A deeper analysis revealed that two of them apparently miss a link to the goals G10 and G01, while one requirement arises directly from the daily tasks performed by the caregivers.

5 Related Work

Research studies in requirements elicitation, and in particular on approaches based on GO modelling, are relevant for our work. First, the comprehensive survey review on empirical research in requirements elicitation by Dieste et al. [4], which derives some conclusions on relative usefulness of different elicitation techniques (e.g. structured interviews gather more information than unstructured interviews; unstructured interviews gather more information than sorting and ranking techniques; and interviewing is cited as the most popular requirements elicitation method). Second, some works define frameworks for the selection of requirements elicitation techniques, within a specific application domain, which propose supporting guidelines for practitioners, as for example [8] and [11]. In addition, [6] defines a general model for an iterative requirements elicitation process, in which the selection of a specific requirements elicitation technique is driven by problem, solution, domain characteristics and the actual requirements set to be consolidated. In fact, for our long term objective we assume as a working hypothesis that the general model proposed by Hickey et al. [6] can be used in practice. This model uses domain characteristics and actual requirements for the selection process, so that we need to find out a way to characterise types of knowledge from them. This is motivating the retrospective analysis described in this paper, since it turns out that the elicitation process adopted in *ACube* can be seen as an instantiation of [6]'s unified model, in which the *ACube* early requirements goal model can represent the actual requirements.

For the specific elicitation techniques exploited in *ACube*, GO approaches applied in real projects in the health care domain such as [2] and, for STS, [1], are worth mentioning, confirming the usefulness of GO modelling to understand such complex domains and to elicit the requirements for STSs in these domains.

6 Discussion and Conclusion

In this paper we described a retrospective analysis of a project aiming at the development of an STS for a social residence for people suffering from Alzheimer's. Findings from a quantitative and qualitative analysis of the available documentation were reported. First, the information sources of the elements in Tropos early requirements model were presented. Among these information sources the domain document and interviews prevailed as the main sources for discovering elements for an early requirements model. Second, concerning the type of knowledge and the corresponding level of abstraction of model elements, knowledge about elements with lower abstraction, namely tasks, were captured mostly from domain documents, while actors and root level goals where mainly derived from domain stakeholder interviews. Moreover, an important number of goals was discovered only during goal modelling, connecting the different abstraction levels and finding the reasons for activities performed. This reveals that a mixed top-down and bottom-up elicitation strategy (as described by Giorgini et al. [5]) was adopted to perform modelling. The analysis shows again that a good documentation is important for keeping a clear understanding of the source of requirements and of the process that was followed by the analysts. Further investigations will be necessary to find missing trace links between the requirements artefacts (e.g., exploiting IR techniques). Moreover, the analysis will be extended to the whole requirements set, including technology-driven and non-functional requirements.

References

1. Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. Requirements Engineering 15(4), 439–458 (2010)
2. An, Y., Dalrymple, P.W., Rogers, M., Gerrity, P., Horkoff, J., Yu, E.S.K.: Collaborative social modeling for designing a patient wellness tracking system in a nurse-managed health care center. In: DESRIST (2009)
3. Cooper, A., Reimann, R., Cronin, D.: About face 3: the essentials of interaction design. Wiley India Pvt. Ltd. (2007)
4. Dieste, O., Juristo Juzgado, N.: Systematic review and aggregation of empirical studies on elicitation techniques. IEEE Trans. Software Eng. 37(2), 283–304 (2011)
5. Giorgini, P., Mylopoulos, J., Perini, A., Susi, A.: The Tropos Methodology and Software Development Environment. In: Yu, Giorgini, Maiden, Mylopoulos (eds.) Social Modeling for Requirements Engineering, pp. 405–423. MIT Press (2010)
6. Hickey, A.M., Davis, A.M.: A unified model of requirements elicitation. J. of Management Information Systems 20(4), 65–84 (2004)
7. Leonardi, C., Sabatucci, L., Susi, A., Zancanaro, M.: Design as Intercultural Dialogue: Coupling Human-Centered Design with Requirement Engineering Methods. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part III. LNCS, vol. 6948, pp. 485–502. Springer, Heidelberg (2011)
8. Maiden, N.A.M., Rugg, G.: ACRE: Selecting Methods For Requirements Acquisition. Software Engineering Journal 11(3), 183–192 (1996)
9. Nuseibeh, B., Easterbrook, S.M.: Requirements engineering: a roadmap. In: ICSE - Future of SE Track, pp. 35–46 (2000)
10. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14(2), 131–164 (2009)
11. Tuunanen, T.: A New Perspective on Requirements Elicitation Methods. The Journal of Information Technology Theory and Application (JITTA) 5(3), 45–72 (2003)

Requirements, Intentions, Goals and Applicable Norms

Alberto Siena[1], Silvia Ingolfo[1], Angelo Susi[3], Ivan J. Jureta[2],
Anna Perini[3], and John Mylopoulos[1]

[1] University of Trento, via Sommarive 14, Trento, Italy
{a.siena,silvia.ingolfo,jm}@unitn.it
[2] University of Namur, 8, rempart de la vierge, 5000 Namur, Belgium
ivan.jureta@fundp.ac.be
[3] FBK-Irst, via Sommarive 18, Trento, Italy
{susi,perini}@fbk.eu

Abstract. Norms such as laws and regulations are an additional source of requirements as they cause domain actors to modify their goals to reach compliance. However, norms can not be modeled directly as goals because of both an ontological difference, and an abstraction gap that causes the need to explore a potentially large space of alternatives. This paper presents the problem of deriving goals from norms and illustrates the open research challenges.

1 Introduction

The requirements elicitation problem is traditionally represented through Zave&Jackson theory $S, D \models R$ [1], in which stakeholder requirements are bound to their specification under the proper domain assumptions. The notion of goal has been widely used to provide a conceptual representation of stakeholder requirements, and to refine them into an operationalization able to translate their intentions into a requirements specification for the system-to-be [2]. This way goals successfully capture the necessary alignment of the specification to stakeholder initial requirements. In recent years, the problem of aligning goals to relevant norms – such as laws and regulations – has grown in importance, complexity, and impact, while business transactions and social activities are increasingly conducted in a global setting. The challenge offered by this context is due to the fact that the nature of norms does not allow to treat them as traditionally done for goals.

For example, a top-level goal such as "communicate with friends" may be or-decomposed into "communicate via email" or "communicate in real-time", which in turn could be decomposed into "chat-based communication" or "mobile communication", and so on. In the end, when the goal tree will be operationalized through tasks, whichever alternative will be chosen it will satisfy the top-level goal. However, if we change the nature of the top-level goal, things may change. A top goal such as "share music with friends" can still be refined and operationalized into a specification that matches the top goal, but a different problem comes into the scene. Intuitively, this goal suggests an intention that can not be acceptable legally: stakeholder requirements can be met, but the involvement of copyright laws delineates a new set of boundaries and restrictions to the achievement of the goal. A problem of a completely different nature

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 195–200, 2012.

is therefore identified, a problem not captured by traditional refinement and operationalization approaches. Goals are a means to provide a representation of the requirements problem, and refinement and operationalization techniques are used to search for a solution to the problem. When a compliance issue exists, norms represent the problem, and goals should be the solution to the compliance problem.

In the present position paper we outline the problem of norm compliance in general, and specifically with respect to goal-oriented requirements engineering. We look at the problem from a conceptual modeling standpoint, where — for the purpose of an easy communication with stakeholders — the complexity of formal reasoning techniques is limited to support model analysis. In this scenario, we claim that the different granularity offered by the concept of situation offers important modeling advantages in both the functional and normative domain. The applicability of norms and their compliance is in fact a dynamic combination that needs to be managed in a flexible and systematic way. Traditional solutions lack the ability to promptly manage this issue and are not always able to offer alternative solutions to the problem at hand. We propose to model the problem with situations [3], a concept underlying and capturing functional and normative aspects (goals, norms, intentions, ...).

The paper is organized as follows: section 2 describes the state of the art; section 3 positions our work by outlining the context and motivation for a relevant research problem; section 4 points out an interesting research direction towards a solution, and discusses the related research challenges; finally, section 5 concludes the paper.

2 Related Work

Norms such as laws and regulations are an additional source of requirements for the system-to-be. Several approaches have been proposed in the literature, to formally tackle this problem using goals. Darimont and Lemoine use KAOS [4] to represent objectives extracted from regulation texts [5] and formalize them using Linear Temporal Logic (LTL). [6] proposes an ontology for achieving coordination purposes in multi-agent systems through the use of commitments. A commitment-based approach is also used in [7] where the proposed methodology guides the requirement engineers in the analysis of commitments, privileges, and rights of policy documents.

Other techniques represent obligations, extracted form regulations, as stakeholders goals. Such techniques have the limitation that they do not have a specific ontology to represent norms, thus allowing only a partial representation of legal prescriptions. For example, Ghanavati et al. [8] use GRL to model goals and actions prescribed by laws, and link stakeholder goals to them through traceability relations. Likewise, Rifaut and Dubois use *i** to model fragments of the Basel II regulation [9]. Worth mentioning that the authors have also experimented this goal-only approach in the Normative *i** framework [10]. That experience focussed on the emergence of implicit knowledge, but the ability to argue about compliance was completely missing, as well as the ability to explore alternative ways to be compliant. We depart from these approaches because we move from the consideration that norms, *per se*, can not be treated as requirements – and represented as goals neither.

Finally it is worth mentioning that there are a number of AI approaches supporting formal reasoning about norms — e.g., default logic, deontic logic and several law-specific logics. Such approaches are largely used in normative Multi-Agent Systems (MAS), in which agents decide their behavior and whether to follow the system's norms, and the normative system specify how agents can modify the norms [11]. We acknowledge such approaches being as better solution for formal reasoning. However requirements engineering involves multiple non-expert stakeholders, and consequently there is a need for higher abstract conceptual modeling techniques, which could support easy knowledge representation and information communication among stakeholders.

3 Problem Statement

When reasoning with norms, a different perspective needs to be adopted with respect to goals. When simple cases are taken in consideration — like the one about music sharing — goal-only approaches can be adopted to ordinarily manage the problem. In reality, scenarios are rarely simple enough to be effectively handled by those methodologies. Norms are generally complex structures, built upon conditions and exceptions, which should be explicitly taken into consideration in order to properly engineer requirements.

Consider for example the self-driving system of an autonomous car. In addition to be technically capable of using sensors and actuators to drive, such systems must fulfill passengers goals while obeying to traffic rules. The (Italian) traffic regulation says for example that it is forbidden to stop the car on the motorway. However, in case of a car failure, it is permitted to stop the car on the motorway, provided that the hazard lights are switched on and passengers wear a reflective safety vest when outside the vehicle. It is clear that in this last case the first standard prohibition is no longer applicable. The car is now exceptionally allowed to stop on the motorway but new obligations have to be met (turning on the hazards lights, and signal the safety vest wearing duty). More cases — like heavy snow or car accident — are handled in other traffic rules and they all provide new possible scenarios as well as new specific rules for the driver. Eventually an actor complies with a norm in a given situation if he satisfies the obligation, prohibition or permission provided by the norm in *that* situation.

We can see how the occurrence of different cases — such as "car failure", "hazard lights are on" — introduces conditional elements that trigger the applicability of different norms. At the same time, the satisfaction of such conditional elements changes the definition of compliance to norms. The intertwinement of the applicability of norms and compliance should therefore take into consideration the role of such conditions, so that it fits the degree of complexity that conditions and exception generate.

Just like conditions happening may be out of the control of an actor's will (e.g., a car failure or heavy fog), the applicable regulations can not be considered *desired* by stakeholders. The notion of goal instead, has in its very heart the purpose and intention to be wanted, desired by an actor. An obligation prescribed by the law (e.g., pay taxes or wear safety vests) has no intention to be desirable. The clear ontological difference between goals and norms, needs to be reflected in the way requirements are treated.

Another important reason why norms can not be represented by goals arises from the abstraction gap between the two concepts. A goal depends on an actor desiring it, and the actor is located in a given specific domain. Norms on the other hand concern *classes* of actors — legal actors, e.g., the driver of a vehicle — and describe the domain with the rights and obligations applying to legal actors in different situations. In short, norms describe classes of actors and their expected behavior by means of rights and duties. These legal actors and their expectations have to be mapped and evaluated in a domain of actors described by their desired goal. Moreover, norms tend to be cross-domain and may apply to multiple situations. Their heterogeneous and sometimes general nature allows regulations to reach and regulate situations of different domains. For example the California smoke-free law applies to all *working places*: hospitals, universities, shopping mall, little bakeries or private offices are all affected and regulated. It is therefore crucial to identify the situations holding both in a specific domain (e.g., "working place") and for a specific actor (e.g., "smoker") in order to evaluate the obligation, prohibition or permission provided by the norm applying in that situation.

The abstraction gap between goals and norms has important implications when it comes to the identification of a solution. In fact, when conditions and exceptions are used to fine-tune legal prescriptions, alternatives arise in the legal space, which need to be explored in the requirement space. This potentially large space of alternatives needs to be carefully and systematically managed to guide and to not confuse requirement engineers. Situations [3] offer the possibility to manage these solutions, as they can describe a state of the world both desired by an actor and addressed by a law. Exploring which situations hold and the applicability of norms based on the situations holding, allow us to identify alternatives in the legal domain. When this (legal) situations are mapped in the functional domain, they are refined and linked to the situations achievable by the actors of a system. The abstraction gap can therefore be managed using the concept of *situation*.

4 Toward a Solution

The problem of dealing with norms and goals lies in goals being at the same time solution to the compliance problem and problem for requirements elicitation. When representing stakeholder requirements, goals act as a guide for identifying a set of requirements specifications that match them. When representing stakeholder behavior, goals must be engineered to match applicable norms. These two different perspectives are not necessarily in conflict. The economic theory of bounded rationality [12] says that actor behavior can be seen as either rational or rule-following [13]. They are rational if they act according to their own interest, after having evaluated alternatives, expectations and preferences. They are rule-following if they choose their behavior on the basis of its appropriateness to the situation, to their identity (the role they play), and to applicable norms. In the bounded rationality theory, actors are partially rational and partially rule-following. The adoption of the bounded rationality approach allows us to model actors normative elements alongside their intentional elements, letting requirements specification emerge on the basis of both types.

This is possible as long as goals and norms can be related with each other. To do this we use the concept of *situation* [3]. Goals are traditionally intended as states of the world, also partially defined, desired by stakeholders. On the other hand norms can be seen as states of the world, also partially defined and imposed to stakeholders. A *situation* is defined as the neutral concept of partial state of the world. When a situation is desired by a stakeholder, it's called *goal*: when the stakeholder achieves his/her goal, a certain situation holds. When a situation is object of a legal taxonomy of choice, it takes different names. It's called *obligation* when the situation must necessarily hold for the stakeholder, it is called *permission* when it is facultative and it may or may not hold for the stakeholder, it is called *prohibition* if the situation must necessarily not hold for the stakeholder. In other words, situation is a neutral concept, which can act as a *lingua franca* between the legal world and the domain.

Research challenges. Eventually, we need to study and understand how the *choice* of a behavior is done, across the many different alternatives. We have identified three problems that should be further investigated: the applicability problem, the appropriateness problem, and the responsibility problem.

Applicability problem. In the rule-following paradigm, actors act on the basis of the norms that hold in the (perceived) situation. They don't have control on this situation. However, actors have goals they want to achieve and alternative goals to choose form: once achieved, goals bring about states of the world which match situations, which in turn may activate some norms (i.e., make them applicable), block them (make them not applicable), comply or violate them, or derogate to others. By selecting one alternative or another, the applicable norms are implicitly chosen as well, and consequently compliance has to be ensured for that specific alternative. Reasoning about which norms apply in a given situation, and how to match such norms in that situation is needed. In [14] we propose a framework for searching in the law variability space, intended as the alternative norms applicable to one or more given situation(s).

Appropriateness problem. After having understood which norms are applicable, the next problem is to evaluate whether a behavior is appropriate to that norms. As said, norms and goals lay at different levels of abstraction. The traditional refinement approach of goal modeling is barely applicable to norms: a norm can affect several goals of several actors, so that *reconciliation* appears to be necessary, rather than refinement. With reconciliation we mean the act of establishing a relation between one (or more) goal to one (or more) norm. Reconciliation in turn can make the complexity explode, because each goal of a goal model could have to be compared to each norm in order to check their semantic alignment.

Responsibility problem. In order to be complied with, norm provisions must be carried out by those actors who are in charge of them. Given a community of actors and a norm requiring one of them to perform a certain action, if the action is correctly performed but the performer is not the actor in charge of it, then the norm is not complied with. In other words, the *identity* of the performer is important as much as the performance itself. A problematic exception to this, is represented by delegation. When a compliance goal is delegated to another actor, who has the capability to operationalize it, there is a discrepancy between the actor who is in charge of complying, and the actor who performs the compliance action. This kind of delegation may or may not be legally acceptable, depending on the delegation right that accompanies the norm.

5 Conclusion

In this position paper we have highlighted the problem of norms such as laws and regulations, and the problem they present in goal modeling. Norms are an additional source of requirements but can not be modeled directly as goals because of both an ontological difference, and an abstraction gap that causes the need to explore a potentially large space of alternatives. Although for simple norms this gap can be unimportant, as the complexity of the legal frame increases, the more ad hoc approaches become necessary. We have highlighted three relevant research problems that attain, respectively, the applicability of norms to goal alternatives, the reconciliation of domain-specific goals with cross-domain norms, and the identification of responsible actors.

References

1. Zave, P., Jackson, M.: Four dark corners of requirements engineering. ACM Trans. Softw. Eng. Methodol. 6(1), 1–30 (1997)
2. Yu, E.S.K., Mylopoulos, J.: Why goal-oriented requirements engineering. In: REFSQ, pp. 15–22 (1998)
3. Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Reasoning with Key Performance Indicators. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 82–96. Springer, Heidelberg (2011)
4. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Science of Computer Programming 20(1-2), 3–50 (1993)
5. Darimont, R., Lemoine, M.: Goal-oriented analysis of regulations. In: Laleau, R., Lemoine, M. (eds.) ReMo2V, held at CAiSE 2006. CEUR Workshop Proceedings, vol. 241. CEUR-WS.org (2006)
6. Singh, M.P.: An ontology for commitments in multiagent systems: Toward a unification of normative concepts. Artificial Intelligence and Law (1999)
7. Young, J.D., Anton, A.I.: A method for identifying software requirements based on policy commitments. In: IEEE Int. Conf. Req. Eng., pp. 47–56 (2010)
8. Ghanavati, S., Amyot, D., Peyton, L.: Towards a Framework for Tracking Legal Compliance in Healthcare. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 218–232. Springer, Heidelberg (2007)
9. Rifaut, A., Dubois, E.: Using goal-oriented requirements engineering for improving the quality of iso/iec 15504 based compliance assessment frameworks. In: Proceedings of RE 2008, pp. 33–42. IEEE Computer Society, Washington, DC (2008)
10. Siena, A., Maiden, N.A.M., Lockerbie, J., Karlsen, K., Perini, A., Susi, A.: Exploring the Effectiveness of Normative *i** Modelling: Results from a Case Study on Food Chain Traceability. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 182–196. Springer, Heidelberg (2008)
11. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multi-agent systems. In: Normative Multiagent Systems 2007 (2007)
12. Simon, H.A.: Models of man: social and rational: mathematical essays on rational human behavior in a social setting. Wiley (1957)
13. March, J.: A Primer on Decision Making: How Decisions Happen. Free Press (1994)
14. Siena, A., Jureta, I., Ingolfo, S., Susi, A., Perini, A., Mylopoulos, J.: Capturing Variability of Law with *Nómos* 2. In: Atzeni, P., Cheung, D., Sudha, R. (eds.) ER 2012. LNCS, vol. 7532, pp. 383–396. Springer, Heidelberg (2012)

Towards Advanced Goal Model Analysis with jUCMNav

Daniel Amyot[1], Azalia Shamsaei[1], Jason Kealey[2], Etienne Tremblay[2],
Andrew Miga[3], Gunter Mussbacher[3], Mohammad Alhaj[3],
Rasha Tawhid[4], Edna Braun[4], and Nick Cartwright[4]

[1] School of EECS, University of Ottawa, Canada
{damyot,asham092}@uottawa.ca
[2] JUCM Software Inc.
{jkealey,etremblay)@jucm.ca
[3] Department of SCE, Carleton University, Canada
andrew_miga@sympatico.ca, gunter@sce.carleton.ca,
malhaj@connect.carleton.ca
[4] Aviation Security Directorate, Transport Canada, Canada
{rasha.tawhid,edna.braun,nick.cartwright}@tc.gc.ca

Abstract. Goal modeling is an important part of various types of activities such as requirements engineering, business management, and compliance assessment. The Goal-oriented Requirement Language is a standard and mature goal modeling language supported by the jUCMNav tool. However, recent applications of GRL to a regulatory context highlighted several analysis issues and limitations whose resolutions are urgent, and also likely applicable to other languages and tools. This paper investigates issues related to the computation of strategy and model differences, the management of complexity and uncertainty, sensitivity analysis, and various domain-specific considerations. For each, a solution is proposed, implemented in jUCMNav, and illustrated through simple examples. These solutions greatly increase the analysis capabilities of GRL and jUCMNav in order to handle real problems.

Keywords: Analysis, Goal-oriented Requirement Language, jUCMNav, strategies, tool support, User Requirements Notation, visualization.

1 Introduction

Goal modeling is an important part of requirements engineering activities. Goal models capture stakeholder and business objectives, alternative means of meeting them, and their positive/negative impacts on various quality aspects. The analysis of such models guides the decision-making process as well as the refinement of imprecise user requirements into precise system requirements.

The Goal-oriented Requirement Language (GRL), part of the User Requirements Notation (URN) [2,5], is a standard notation for goal modeling. GRL enables requirements engineers and business analysts to describe stakeholders (actors) and intentions (e.g., goals, softgoals, and tasks), together with their decomposition structure, dependencies, and contribution levels. Given initial satisfaction levels associated with

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 201–210, 2012.

some of the elements of a goal model (i.e., a strategy), tool-supported analysis techniques can determine the satisfaction levels of the other elements [1]. In particular, jUCMNav [3, 6] is a free Eclipse plug-in that enables the creation and management of complex GRL models. It also provides features to support various analysis algorithms that exploit strategies, to help visualize analysis results, and to generate reports.

Yet, the realities of complex application domains, such as regulatory compliance [8], have pushed the limits of the language and of current tool support. Through our experience modeling and analyzing real regulations with GRL, we have observed important issues related to the comparison of strategies and evolving models, the management of complexity of sets of strategies, the management of uncertainty related to contribution links, the sensitivity of analysis results when localized changes are explored, the usability of the standard GRL evaluation scale, the practicality of unilingual models in a multilingual environment, and facilities for handling strategies separately from their model.

This paper explains each of these issues and proposes solutions that we have implemented in the latest version of the jUCMNav tool, with simple but illustrative examples. We believe these solutions will help address similar issues beyond the regulatory compliance context. They may also inspire language designers to evolve other goal-oriented languages and their tools.

2 Strategy and Model Differences

In GRL (see metamodel extract in Fig. 1), a model can include *evaluation strategies*, which are sets of initial *evaluations* (quantitative value or qualitative labels) associated with *intentional elements* [5]. Strategies are also *grouped* for classification and convenience. Various qualitative, quantitative and hybrid propagation algorithms take these values and propagate them to the other intentional elements (through contribution, decomposition and dependency links), and to actors that contain intentional elements with non-null importance [1]. In GRL, the importance level of an intentional element to its actor is shown between parentheses (e.g., see Fig. 2). Intuitively, using a quantitative scale (as used in our examples), the satisfaction level of an intentional element is: the maximum of the children's evaluation values for an OR decomposition, the minimum for an AND decomposition, and the bounded weighted sum for contributions. jUCMNav also uses color feedback to highlight satisfaction levels (the greener the better, the redder the worse) as well as dashed lines for the border of intentional elements that are part of strategies (see Fig. 2).

Usually, many strategies are defined for a model to explore different global alternatives or tradeoffs in a decision support context, to represent as-is and to-be contexts, or to capture historical contexts (e.g., the situation or compliance level of the organization at different times). There is a need to compare strategies and to visualize this comparison in terms that the model user can understand. jUCMNav already supports the generation of reports (in PDF, RTF, and HTML formats) that contain a tabular representation of all strategies and their results. This is useful for sharing models and strategy evaluations with people who do not have access to the modeling tool, but this is not really amenable to the real-time analysis of differences between strategy results. The issue here is: *Can we highlight differences within the graphical model itself in order to provide more immediate feedback and support discussions between stakeholders around the model, its strategies, and the supporting tool?*

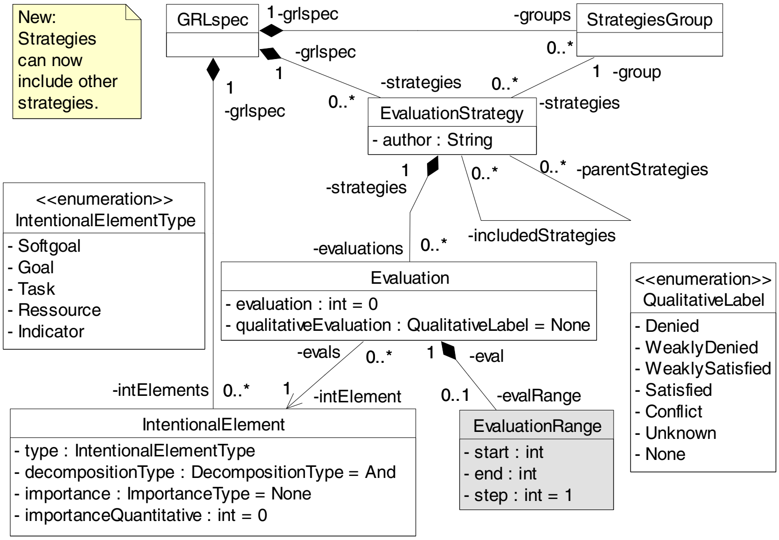
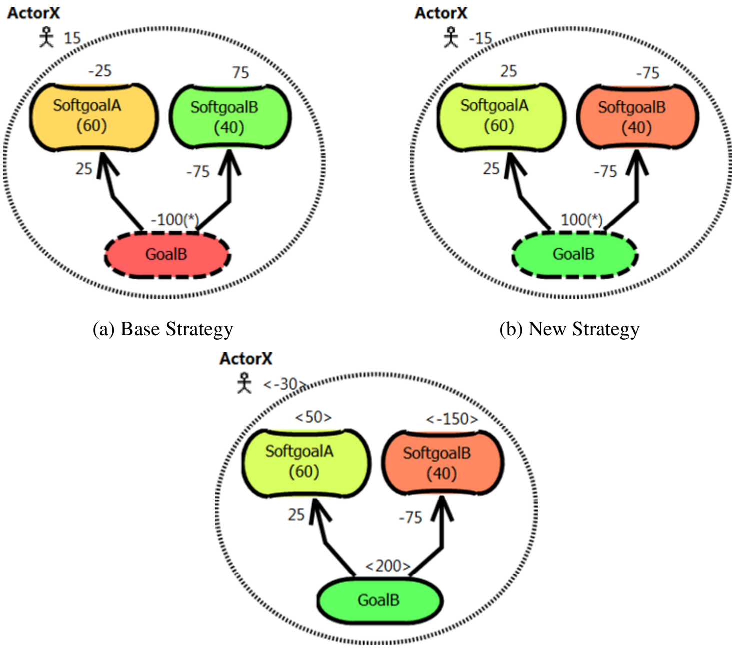


Fig. 1. Extract of URN metamodel – Strategies

To answer this question, we propose a new jUCMNav feature that highlights *strategy differences* visually in terms of evaluations of intentional elements and actors. The difference is computed between a base strategy (e.g., Fig. 2a) and a current strategy (e.g., Fig. 2b) on a per element basis (including actors). The standard GRL scale for quantitative evaluations goes from –100 (fully denied, shown in red) to 0 (neutral, in yellow) to +100 (fully satisfied). Consequently, the difference scale is [–200..200]. Differences are displayed between angle brackets (to differentiate them from normal satisfaction values), again with color feedback (<–200> in red, <0> in yellow, and <+200> in green), so the tradeoffs can be understood at a glance. Fig. 2c shows the difference results of our simple example; with the new strategy, ActorX becomes less satisfied by a difference of 30. jUCMNav allows one to select a base strategy and then switch between many alternative strategies to visualize (instantly) their differences.

In a context where the GRL models themselves and their strategy definitions evolve (e.g., as we gain more insights about the domain being modeled), another question is: *How can we highlight, understand, and control model evolution?* Ideally, *model differences* would need to be done at the level of GRL graphical model elements. However, this poses technical challenges, especially for the presentation of deletions and modifications of model elements and their properties.

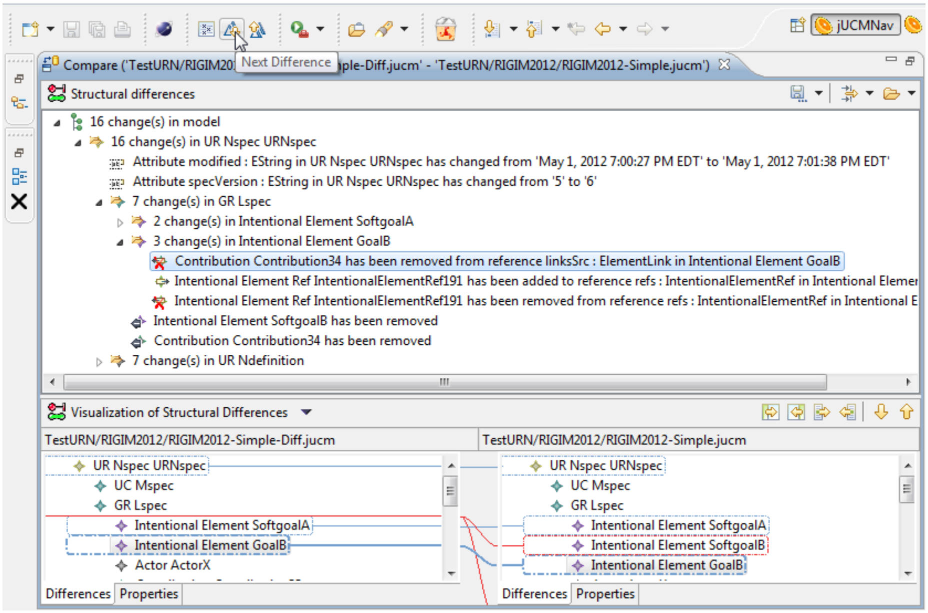


(a) Base Strategy

(b) New Strategy

(c) Difference: New Strategy – Base Strategy

Fig. 2. Strategy difference example



Compare ('TestURN/RIGIM20 ple-Diff.jucm' - 'TestURN/RIGIM2012/RIGIM2012-Simple.jucm')

Structural differences

- 16 change(s) in model
  - 16 change(s) in UR Nspec URNspec
    - Attribute modified : EString in UR Nspec URNspec has changed from 'May 1, 2012 7:00:27 PM EDT' to 'May 1, 2012 7:01:38 PM EDT'
    - Attribute specVersion : EString in UR Nspec URNspec has changed from '5' to '6'
    - 7 change(s) in GR Lspec
      - 2 change(s) in Intentional Element SoftgoalA
      - 3 change(s) in Intentional Element GoalB
        - Contribution Contribution34 has been removed from reference linksSrc : ElementLink in Intentional Element GoalB
        - Intentional Element Ref IntentionalElementRef191 has been added to reference refs : IntentionalElementRef in Intentional Elemer
        - Intentional Element Ref IntentionalElementRef191 has been removed from reference refs : IntentionalElementRef in Intentional E
      - Intentional Element SoftgoalB has been removed
      - Contribution Contribution34 has been removed
    - 7 change(s) in UR Ndefinition

Visualization of Structural Differences

TestURN/RIGIM2012/RIGIM2012-Simple-Diff.jucm	TestURN/RIGIM2012/RIGIM2012-Simple.jucm
UR Nspec URNspec	UR Nspec URNspec
UC Mspec	UC Mspec
GR Lspec	GR Lspec
Intentional Element SoftgoalA	Intentional Element SoftgoalA
Intentional Element GoalB	Intentional Element SoftgoalB
Actor ActorX	Intentional Element GoalB

Differences | Properties

Fig. 3. URN model difference in jUCMNav based on EMF Compare

The approach we have prototyped in jUCMNav reuses the facilities of the *EMF Compare* plugin [4], a generic difference engine for modeling tools based on the Eclipse Modeling Framework (EMF). EMF Compare represents a simple and yet efficient solution to the comparison of URN/GRL models. For example, Fig. 3 displays the results of comparing the simple model used in Fig. 2 with one where we have removed SoftgoalB (including its incoming contribution), and changed some attributes. EMF Compare also allows one to copy changes (or merge) from one version to the other. Finally, EMF Compare offers means to filter out comparison results of little value (e.g., a change in the size or position of an element) in order to focus on the most important changes. However, filtering is left for future work in our context.

3 Complexity/Uncertainty Management and Sensitivity Analysis

Complexity in goal models can take many forms. One is related to the size of the models and the number of strategies to handle. jUCMNav already offers several features to handle models that include many diagrams (e.g., navigation, search, different views, and the sorting of diagrams). However, one issue remains: *How should we manage large collections of strategies?*

Our solution is to have a parent-child *inclusion relationship* between strategies (see the corresponding new association in Fig. 1). In essence, a parent strategy can now include another strategy, which means that the initial evaluations of the latter will be included automatically (i.e., reused) in those of the former. These included evaluations can then be overridden by parent evaluations (if they target the same intentional element), or complemented by additional evaluations. Strategy inclusion can be done recursively (across many levels). jUCMNav ensures that inclusion loops are avoided. This solution hence improves consistency and reduces the number of updates required when new strategies or model elements are added. It can also be combined with the strategy difference feature described in the previous section.

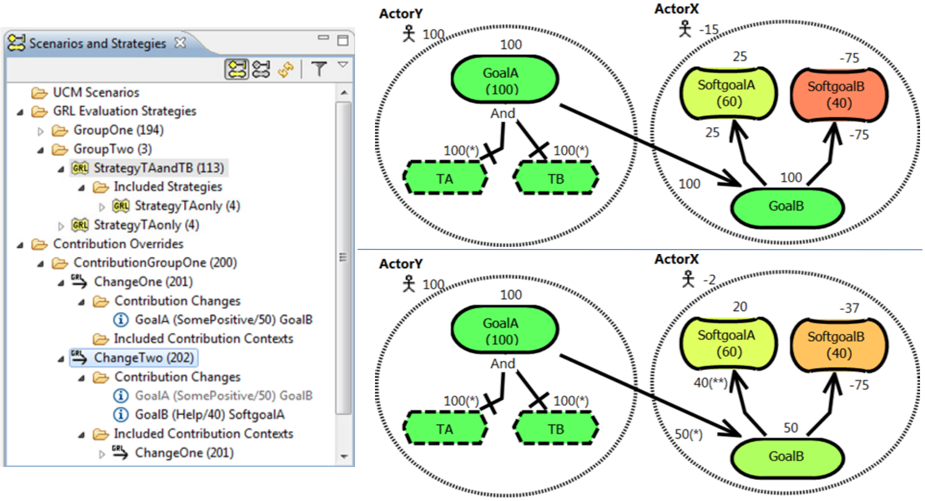


Fig. 4. Examples of strategy inclusions and of contribution contexts

As an example, the model in Fig. 4 (top) was evaluated against StrategyTAandTB (selected in the left view), which includes StrategyTAonly (initializing the evaluation of task TA with 100) and adds as a second evaluation, this time for task TB (100).

Another level of complexity lies in the uncertainty surrounding weights (or levels) of contribution links in goal models. It is often difficult for modelers to determine appropriate contribution levels for such links (see the Despair sin in [7]), and the real impact of using different levels is difficult to assess. The issue here becomes: *Can we investigate alternative combinations of contribution levels during analysis without having to produce and maintain different variants of a goal model?*

Our proposed solution is to support the concept of *contribution contexts*, which are to contribution levels what strategies are to evaluation values. As formalized in Fig. 5 (the dark gray metaclasses are new), a contribution context contains a set of *contribution changes*, which override the quantitative and/or qualitative contribution levels of contribution links in a GRL model. Like for strategies, contribution contexts are *grouped*, they can include other contexts, and they can be used in strategy differences.

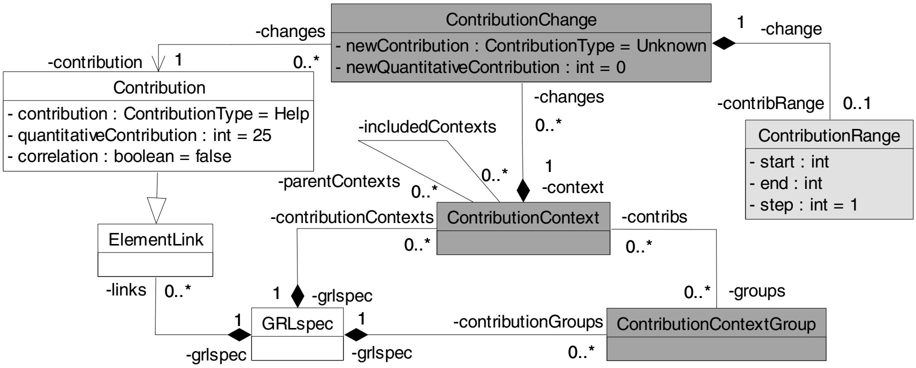


Fig. 5. Extract of URN metamodel – Contribution contexts

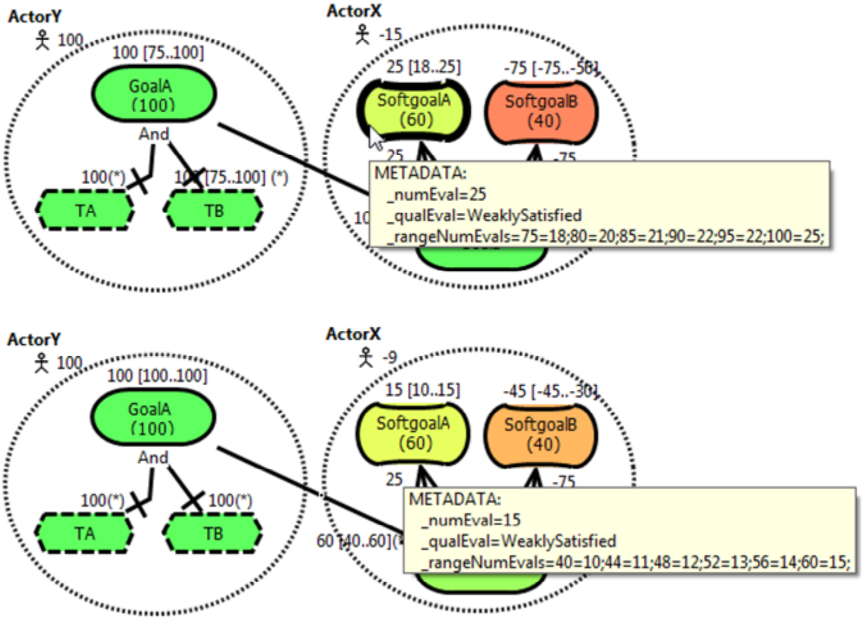
In the left view of Fig. 4, ChangeOne changes the contribution from GoalA to GoalB to 50, whereas ChangeTwo includes ChangeOne *and* overrides the contribution from GoalB to SoftgoalA with 40. The result of evaluating StrategyTAandTB against the model modified by ChangeTwo is shown at the bottom of Fig. 4. Note that (**) on a contribution indicates a direct change while (*) indicates an included change.

A third issue that touches both complexity and uncertainty is whether localized changes to a satisfaction level or to a contribution level actually impact significantly or not the satisfaction of high-level objectives in a goal model. This is akin to sensitivity analysis, which is the study of how the variation (uncertainty) in the output of a model can be attributed to different variations in the inputs of the model. The problem is as follows: *Can we support simple sensitivity analysis without having to declare strategies and contribution contexts for all single values of interest?*

Our proposed solution is to allow for *ranges* of values to be used for strategy evaluations (Evaluation Range in Fig. 1) and for contribution changes (ContributionRange in Fig. 5) instead of just single values. The *step* of a range is the increment by which we iterate from the *start* to the *end*. By associating a range to an initial evaluation, all other intentional elements impacted directly or indirectly will also have a range, but this time for computed values. Fig. 6 (top) shows an example where TB has an initial range of [75..100] with 5 as a step value. TA is not impacted, but all of the other intentional elements are. Their resulting ranges are also displayed. In addition, all intermediate values (for each iteration) are accessible as metadata, and hence visible as a tooltip by hovering over the desired element (SoftgoalA in Fig. 6). This simple sensitivity analysis enables the modeler or analyst to assess the impact of localized changes and to determine whether a change to an initial satisfaction value really matters or not.

A similar usage is possible for contributions. Fig. 6 (bottom) shows an example where the contribution from GoalA to GoalB is overridden by a [40..60] range with a step of 4. The results are shown for StrategyTAandTB, which does not include any evaluation range. Again, the impact on intentional elements can easily be assessed.

Sensitivity analysis in jUCMNav is currently limited to one dimension only, i.e., to a range for one evaluation or for one contribution. Allowing for more than one dimension to be explored at once would lead to visualization challenges (e.g., tables or cubes instead of linear arrays of values) that would negatively impact understanding. Other visualization schemes are required in that context. The support for ranges on the actors and possibly on importance values is also left to (near) future work.



```
METADATA:
  _numEval=25
  _qualEval=WeaklySatisfied
  _rangeNumEvals=75=18;80=20;85=21;90=22;95=22;100=25;
```

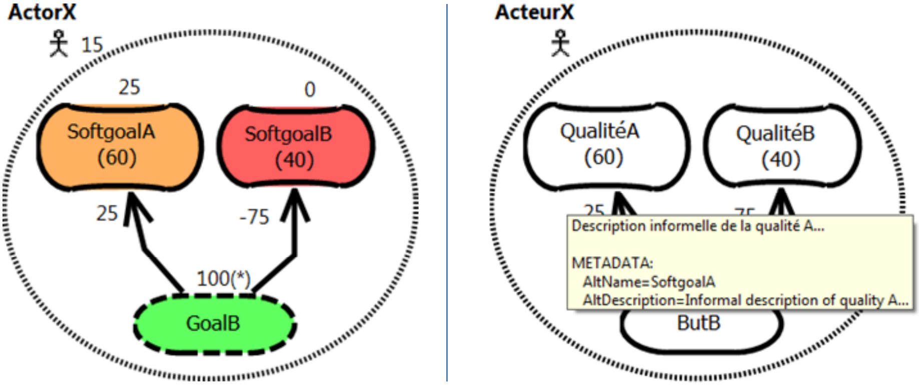
```
METADATA:
  _numEval=15
  _qualEval=WeaklySatisfied
  _rangeNumEvals=40=10;44=11;48=12;52=13;56=14;60=15;
```

Fig. 6. Use of ranges for sensitivity analysis in strategy evaluations (top) and in contributions levels (bottom)

4 Domain Considerations during Analysis

While interacting with policy makers and other stakeholders, we realized that the standard GRL satisfaction range ([–100..100]) was really counter-intuitive to many people, even more so when a goal with a negative evaluation that has a negative contribution to another intentional element leads to a positive evaluation value for that element (see Fig. 2a). This issue was also raised by many undergraduate and graduate students to whom GRL was taught over the past 8 years. This problem is therefore stated as: *Can we support an alternative range of satisfaction values for domains where the standard one is counter-intuitive?*

We have implemented an alternative [0..100] evaluation scale (where 0 now means fully denied) and adapted the user interface (e.g., pop-up menus with predefined values) and the propagation algorithms accordingly. The color feedback in jUCMNav now also depends on the scale being used (with the new scale, 0 is red as there is no longer any negative satisfaction values, and 50 is yellow). Fig. 7 (left) shows the same model and strategy as in Fig. 2b, but evaluated with the new scale. Note that a satisfaction level of 25 is orange now, indicating partial dissatisfaction, rather than light green. Contributions are still allowed to be negative, but they cannot lead to a negative satisfaction values; this is why the evaluation value of SoftgoalB is 0, i.e., the lowest value allowed by this new scale. The modeler can choose between one scale or the other when creating a model. After a few weeks of usage and the training of nearly 50 people in the government on GRL for regulations, there is much ad hoc evidence that this indeed leads to a more intuitive interpretation (especially by non-experts in GRL) of goal models used for compliance analysis.



Description informelle de la qualité A...

METADATA:
AltName=SoftgoalA
AltDescription=Informal description of quality A...

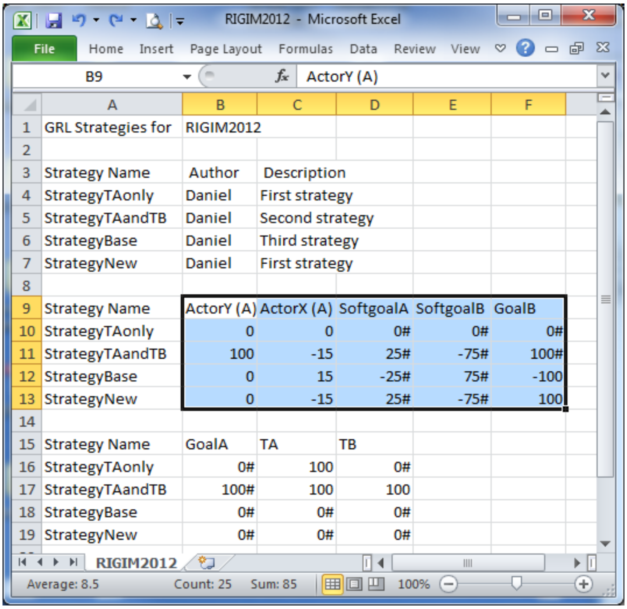
Fig. 7. Strategy evaluation in a [0..100] scale (left) and multilingual model (right)

Another interesting domain consideration is that in Canada, regulations are written in two languages (English and French). Obviously, creating French and English versions of a same model in not desirable. The issue here is: *Can we support models in multiple languages without having different models, to avoid maintenance issues?*

jUCMNav's user interface is already multilingual (and supports French and English), but this is sufficient as there is no way of attaching multiple names and descriptions to model elements. We implemented a feature that allows the modeler to switch between model languages and to provide alternative names and descriptions for model elements, including actors, goals, strategies, and diagrams. When switching languages, the name and description of each element are swapped with alternative values attached to the element as metadata. This is limited to two languages at the moment, but this could be extended to more than two in the future. Fig. 7 (right) presents the French version of the names and descriptions used in Fig. 7 (left). Both are stored in the same model and hence can be easily maintained as the model evolves (minimizing the risk of inconsistencies). There is no automatic translation at the moment as this was seen as potentially dangerous in a regulatory context, but this could likely be added in the future. The same feature is also being explored to support many levels of language in the same context (e.g., for regulation experts, and for non-experts).

One last interesting domain issue that we considered as part of our recent work relates to the fact that, sometimes, strategies need to be stored independently from models. In a compliance context, the people creating a GRL model may not have sufficient privileges to access strategies used to evaluate the model. For example, analyzing the impact of airport incidents might require access to highly confidential data used to populate initial values in the strategies. Moreover, strategies might be generated automatically from data sources (e.g., airport inspection reports) and their results consumed by other analysis and reporting tools (e.g., for Business Intelligence). Hence: *Can we handle strategies and their results separately from their GRL model?*

Our solution involves the *import/export of strategies*, with results, as simple comma-separated value (CSV) files. This enables one to split strategy definitions and results from the model, and hence they can be stored in different places and be restricted to particular users. This format is also easy to process as output (e.g., from a database, or from Microsoft Excel as seen in Fig. 8) or as input (e.g., to a business intelligence tool, or to Excel). Rows represent named strategies while columns represent mainly the actors (results only) and intentional elements. One particularity is that we separate, for intentional elements, results (suffixed with the # symbol, which can be removed easily for post-processing when needed) from definitions (no # symbol).



	A	B	C	D	E	F
1	GRL Strategies for	RIGIM2012				
2						
3	Strategy Name	Author	Description			
4	StrategyTAonly	Daniel	First strategy			
5	StrategyTAandTB	Daniel	Second strategy			
6	StrategyBase	Daniel	Third strategy			
7	StrategyNew	Daniel	First strategy			
8						
9	Strategy Name	ActorY (A)	ActorX (A)	SoftgoalA	SoftgoalB	GoalB
10	StrategyTAonly	0	0	0#	0#	0#
11	StrategyTAandTB	100	-15	25#	-75#	100#
12	StrategyBase	0	15	-25#	75#	-100
13	StrategyNew	0	-15	25#	-75#	100
14						
15	Strategy Name	GoalA	TA	TB		
16	StrategyTAonly	0#	100	0#		
17	StrategyTAandTB	100#	100	100		
18	StrategyBase	0#	0#	0#		
19	StrategyNew	0#	0#	0#		

Fig. 8. Strategies (definitions and results) as imported/exported CSV files

For example, GoalB is initialized with –100 in StrategyBase, but SoftgoalB is not initialized. Another feature is that, when there are many intentional elements, the format allows for a user-defined number of columns to be used, which is convenient for inputs from tools such as Excel (as less horizontal scrolling is required). Strategies then span multiple rows.

During an import, jUCMNav currently creates a new strategy group where the imported strategy definitions are stored (results with a # and actor evaluations are simply ignored). This allows for multiple versions of the strategies (e.g., compliance results evaluated at different times) to be used and then compared. Future work items for this mechanism include the support for strategy groups, included strategies, and ranges.

5 Conclusions

This paper presented many concrete issues with the applicability of goal modeling, and particularly of GRL and jUCMNav, for supporting analysis in a real context. We proposed and implemented a collection of advanced analysis and management features to handle these issues. Although these features represent major advancements over past jUCMNav versions [3], many remaining items for future work have been identified. The real usefulness and validity of these new features also requires further experiment. Regulatory compliance was used here as a context but we suspect that the identified issues and proposed solutions will also be valid for other domains, and probably even for other languages and tools. We finally plan to propose some of our language extensions to become part of a future release of the URN standard [5].

References

1. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.: Evaluating Goal Models within the Goal-oriented Requirement Language. International Journal of Intelligent Systems 25(8), 841–877 (2010)
2. Amyot, D., Mussbacher, G.: User Requirements Notation: The First Ten Years, The Next Ten Years. Journal of Software (JSW) 6(5), 747–768 (2011)
3. Amyot, D., Mussbacher, G., Ghanavati, S., Kealey, J.: GRL Modeling and Analysis with jUCMNav. In: 5th Int. i* Workshop (iStar 2011), Trento, Italy. CEUR-WS, vol. 766, pp. 160–162 (August 2011)
4. Eclipse Foundation: EMF Compare (2012), `http://www.eclipse.org/emf/compare/`
5. International Telecommunication Union: Recommendation Z.151 (11/08), User Requirements Notation (URN) – Language definition (2008), `http://www.itu.int/rec/T-REC-Z.151/en`
6. jUCMNav, Version 5.1.0, University of Ottawa, `http://softwareengineering.ca/jucmnav`
7. Mussbacher, G., Amyot, D., Heymans, P.: Eight Deadly Sins of GRL. In: 5th Int. i* Workshop (iStar 2011), Trento, Italy. CEUR-WS, vol. 766, pp. 2–7 (August 2011)
8. Tawhid, R., Alhaj, M., Mussbacher, G., Braun, E., Cartwright, N., Shamsaei, A., Amyot, D., Behnam, S.A., Richards, G.: Towards Outcome-Based Regulatory Compliance in Aviation Security. In: Requirements Engineering (RE 2012). IEEE CS, USA (to apppear, 2012)

Sixth International Workshop on Semantic and Conceptual Issues in GIS (SeCoGIS 2012)

Preface

Recent advances in information technology have changed the way geographical data were originally produced and made available. Nowadays, the use of Geographic Information Systems (GIS) is not reserved anymore to the specialized user. GISs are emerging as a common information infrastructure, which penetrates into more and more aspects of our society. The technological drift implies a profound change in mentality, with a deep impact on the way geographical data needs to be conceptualized. New methodological and data engineering challenges must be confronted by GIS researchers in the near future in order to accommodate new users' requirements for new applications.

The SeCoGIS workshop intends to bring together researchers, developers, users, and practitioners with an interest in all semantic and conceptual issues in GISs. The aim is to stimulate discussions on the integration of conceptual modeling and semantics into various web applications dealing with spatio-temporally referenced data and how this benefits end-users. The workshop provides a forum for original research contributions and practical experiences of conceptual modeling and semantic web technologies for GIS, fostering interdisciplinary discussions in all aspects of these two fields and highlighting future trends in this area. The workshop is organized in a way to stimulate interaction amongst the participants.

This edition of the workshop received much more submissions than previous editions, 25 submissions, from which the Program Committee selected 10 high quality papers, corresponding to an acceptance rate of 40%. The authors of the accepted papers are world-wide distributed, making SeCoGIS a truly international workshop. The accepted papers were organized in four sessions. The first one contains a keynote speaker and one accepted paper. The next ones contain three papers each. The second session is about semantic issues of geographic data modeling. The third one is about conceptual modeling of geographic applications. The fourth one is about a few more technical aspects of spatio-temporal data modeling.

We would like to express our gratitude to the Program Committee members for their qualified work in reviewing papers, the authors for considering SeCoGIS as a forum to publish their research, and the ER 2012 organizers for all their support.

June 2012

Eliseo Clementini
Esteban Zimanyi
Program Co-Chairs
SeCoGIS 2012

Key Ingredients for Your Next Semantics Elevator Talk

Krzysztof Janowicz[1] and Pascal Hitzler[2]

[1] Department of Geography, University of California, Santa Barbara, USA
jano@geog.ucsb.edu
[2] Kno.e.sis Center, Wright State University, Dayton, OH
pascal.hitzler@wright.edu

Abstract. 2012 brought a major change to the semantics research community. Discussions on the use and benefits of semantic technologies are shifting away from the *why* to the *how*. Surprisingly this more in stakeholder interest is not accompanied by a more detailed understanding of *what* semantics research is about. Instead of blaming others for their (wrong) expectations, we need to learn how to emphasize the paradigm shift proposed by semantics research while abstracting from technical details and advocate the added value in a way that relates to the immediate needs of individual stakeholders without overselling. This paper highlights some of the major ingredients to prepare your next *Semantics Elevator Talk*.

Keywords: Semantics, Ontology, Linked Data, Interoperability.

1 Introduction

Recently, we came across a Gartner Hype Cycle from 2006. It showed the term *Public Semantic Web* as currently entering the bottom of the *Trough of Disillusionment*, while *Corporate Semantic Web* was approaching the earlier *Peak of Inflated Expectations*. The Semantic Web community and related disciplines were questioning whether the field would recover or vanish. The Gartner picture made a dry statement: *5 to 10 years to mainstream adoption*. At hindsight, it seems amazing how profoundly accurate the forecast has turned out to be. Indeed, six years later, Steve Hamby announced 2012 as *The Year of the Semantic Web* in his Huffington Post article by listing a number of highly visible and prominent adoptions including Google's Knowledge Graph, Apple's Siri, Schema.org as cooperation between Microsoft, Google, and Yahoo!, Best Buy Linked Data, and so forth.[1] One could easily add more success stories for semantic technologies and ontologies such as the Facebook Open Graph protocol, The New York Times Web presence, or IBM's Watson system, and still just cover the tip of the iceberg.

[1] See http://www.huffingtonpost.com/steve-hamby/semantic-web-technology_b_1228883.html and www.huffingtonpost.com/steve-hamby/2012-the-year-of-the-sema_b_1559767.html

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 213–220, 2012.

While we see mainstream adoption in industry, academia, and governments, semantics research is far from over. Key research questions have yet to be solved and the wide adoption of more complex semantic technologies and of knowledge engineering is a distant goal on the horizon. Often, past research has provided conceptual insights and purely theoretical approaches to pressing topics such as how to address semantic interoperability, but failed to deliver ready-made solutions. As a research community, we are suddenly faced with discussions shifting away from the *why* to the *how*. Our technical language, loaded with the infamous three-letter acronyms, is not suitable to explain the immediate added value of adopting semantic technologies to stakeholders. With the dawning data revolution, the Semantic Web community is confronted with the need to provide working solutions for data publishing, retrieval, reuse, and integration in highly heterogeneous environments. Interdisciplinary science and knowledge infrastructures such as NSF's Earthcube[2] are among the most promising areas to put semantics to work and to show the immediate added value of our research [1].

Targeting the semantics research community, this paper highlights some of the ingredients required to prepare a semantics elevator talk that explains the value proposition of the Semantic Web to interdisciplinary scientists and at the same time circumnavigates common misunderstandings about the adoption of semantic technologies.

2 The Value Proposition of the Semantic Web

What can be achieved by using the Semantic Web that was not possible before is among the most frequent questions raised when introducing the Semantic Web to stakeholders, and *nothing* is probably the most honest answer. Instead, and more appropriately, one should ask whether a certain project would be realized *at all* without the aid of semantic technologies – in other words, the question is not what is doable, but what is *feasible*. In the following, we list three examples that demonstrate the added value of semantics in different stages of scientific workflows, and which are driven by the immediate needs of scientists instead of abstract assertions.

2.1 Publishing and Retrieving

Participating in the Semantic Web is a staged process and the entry level has been constantly lowered over the past few years, thereby contributing to the success of Linked Data [2] in science, governments, and industry. For the individual scientist, the added value of semantic technologies and ontologies starts with publishing own data. By creating more intelligent metadata, researchers can support the discovery and reuse of their data as well as improve the reproducibility of scientific results. This aspect is increasingly important as journals and conferences ask authors to submit their data along with the manuscripts.

[2] http://earthcube.ning.com/

Semantically annotated data also enables search beyond simple keyword matching. Google's *things not strings* slogan implemented in their new Knowledge Graph shows semantic search in action and highlights how single pieces of data are combined and interlinked flexibly.[3] In a scientific context and combined with Big Data, semantic search and querying will go further and allow to answer complex scientific questions that span over scientific disciplines [1]. With EarthCube, NSF is currently establishing such an integrated data and service infrastructure across the geosciences. New semantics-enabled geographic information retrieval paradigms employ ontologies to assist users in browsing and discovering data based on analogies and similarity reasoning [3,4,5]. To give concrete examples, the paradigm shift from data silos to interlinked and open data will support scientists in searching for appropriate study areas, in finding data sources which offer a different perspective on the same studied phenomena to gain a more holistic view, and in interlinking their own data with external datasets instead of maintaining local and aging copies.

2.2 Interacting and Accessing

One of the key paradigm shifts proposed by the Semantic Web is to enable the creation of smart data in contrast to smart applications. Instead of developing increasingly complex software, the so-called business logic should be moved to the (meta)data. The rationale is that smart data will make all future applications more usable, flexible, and robust, while smarter applications fail to improve data along the same dimensions. To give a concrete example, faceted search interfaces and semantics-enabled Web portals can be created with a minimum of human interaction by generating the facets via the roles and their fillers from the ontologies used to semantically annotate the data at hand. Changes in the underlying ontologies and the used data are automatically reflected in the user interface. In fact, users can even select their preferred Linked Data browser as along as the data is available via a SPARQL endpoint. One example for such a semantics-enabled portal that is semi-automatically generated out of ontologies and data is the Spatial Decision Support portal [6]. In terms of added value, semantic technologies and ontologies reduce implementation and maintenance costs and enable users to access external datasets via their preferred interface, thus benefiting data publishers and consumers. Due to the high degree of standardization and reasoning capabilities enabled by the formal semantics of knowledge representation languages, most available Semantic Web software is compatible. For instance, data can be easily moved between triple stores.

2.3 Reusing and Integrating

Semantic technologies and ontologies support horizontal and vertical workflows, i.e., they offer approaches for all phases starting from data publishing, sharing,

[3] http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

discovery, and reuse, to the integration of data, models, and services in heterogeneous environments. For many scientists and engineers, the reuse and integration aspects may be those with the clearest added value, as 60% of their time is spent on making data and models compatible [7]. By restricting the interpretation of domain vocabularies towards their intended meaning, ontologies reduce the risk of combining unsuitable data and models. A purely syntactic approach or natural language descriptions often fail to uncover hidden incompatibilities and may result in misleading or even wrong results [8].

However, improving semantic interoperability is not the only added value with respect to data reuse and integration. Semantic technologies also support the creation of rules for integrity constraint checking. To give a concrete example, a scientist may import vector data on afforested areas into a semantics-enabled Geographic Information System that checks the data against a selected ontology to display those areas that correspond to a specific *Forest* definition [9]. Finally, semantic technologies and ontologies can also assist scientists in selecting appropriate analysis methods, e.g., by verifying that a particular statistics returns meaningful results when applied to the dataset at hand.

3 Adoption Steps

For potential adopters of semantic technologies, it is often important that rapid progress is made which quickly leads to visible and testable added value. This aspect should not be underestimated. Adopters need to justify their investments, and it could be perceived as a high risk approach if benefits were a long time coming. At the same time, the powerful added value of adopting semantic technologies only unfolds in full in later stages of adoption. The challenge is, thus, to keep the ball rolling through the early adoption stages, such that the greater benefits can be reaped in the medium and long term. The need for rapid adoption can be met with semantic technologies, however a certain minimum of care needs to be taken to make sure that adoption reaches the later and even more beneficial stages. In this section, we point out some key issues related to this staged adoption.

At first, however, it is important for adopters to realize that some semantic technologies have a steep learning curve, and, similarly to engineering disciplines, require a certain routine. Adopters will need an infusion of expert knowledge, either by hiring semantic technology experts or by closely cooperating with them. These experts should be honest about the limits of certain technologies and willing to listen to domain and application problems instead of approaching them with domain-independent blueprints. The Semantic Web is extremely rich, there is always more than one way to go. However, this also requires that potential adopters communicate their needs and ask about the pros and cons of available options. All these problems are well known from working in interdisciplinary teams and, at its core, semantics is all about heterogeneity.

3.1 Rapid Initial Adoption

Rapid adoption starts with publishing data following the Linked Data paradigm. In essence, this means making the data available in a standardized and simple syntactic format, namely in RDF [10]. It is important to understand that this first step does not necessarily add any relevant semantics to the data.

Immediate benefits for the adopter include the following.

– Stakeholders can find the data and access it with common tools which can handle RDF and the RDF semantics. Hence, the barrier to find and reuse data is lowered considerably.
– The adopter's data will become part of the active research community which is concerned with analyzing, understanding, improving, interlinking, and using Linked Data for various purposes.
– Data can be combined with external data via links without the need to keep local copies of such external datasets.
– The adopter gains visibility and reputation by contributing to an open culture of data and as part of the state-of-the-art Linked Data effort.

With those benefits in mind, it is also important to point out what Linked Data does *not* deliver [11,12,13].

– A common syntax helps to lower the barrier for reuse, but does not address semantic interoperability nor does it enable complex queries across datasets, which means that data curation is still a major and non-trivial effort. Essentially, data that is published using informal or semi-formal vocabularies is still wide open to ambiguities and misinterpretations. While this may be less problematic for interaction with human users, it sets clear limits for software agents.
– The *links* in Linked Data are often created ad-hoc with a more-is-better mentality instead of strategies to assess quality, or to maintain and curate already established links. Indeed, many of those links are `owl:sameAs` links which, however, are usually not meant to carry the formal semantics they would inherit from the Web Ontology Language OWL [11,14].
– The paradigm shift to triples as units of meaning and URIs as global identifiers alone is not sufficient to contribute to the Linked Data cloud. A set of methods and tools is required [15]. As research community we have to provide best practice and strategies for different types of stakeholders and projects.

Summing up, publishing Linked Data is a major first step and offers immediate added value at low cost (in terms of time and infrastructure). This step alone, however, does not automatically enable many of the promises of the Semantic Web. In fact, many of the early Linked Data projects merely ended up as more data [13].

3.2 Medium- and Long-Term Bootstrapping

In order to understand how to initiate a medium- and long-term process in adopting *deep* semantic technologies, let us first dwell on one of the key fallacies to adopting semantics in a *rapid* fashion. As pointed out above, such a rapid adoption essentially establishes a common syntax and otherwise relies on the use of vocabularies whose meaning is usually not formally defined and requires substantial human interaction and interpretation.

To make a very simple example for potential difficulties, consider the ad-hoc vocabulary term `ex:hasEmail`, informally described as an RDF propoerty having as values strings which are email addresses of contact persons of a particular nature preserve. Now assume that some of these contact persons use a common email account, e.g., to share responsibilities. Usually, this does not cause any difficulties and is, in fact, common practice. However, a knowledge engineer may, at some later stage, be in need of having more powerful semantics at hand, e.g., because on the Web email addresses are often used as identifiers for account holders, and thus it seems reasonable to assume that `ex:hasEmail` is an inverse functional property in the exact sense in which OWL specifies it.[4] Regretfully, it turns out that this apparently harmless strengthening of the semantics of the vocabulary term `ex:hasEmail` now yields undesired consequences. According to the OWL semantics, we can now conclude that all contact persons having the same email address are, in fact, identical (in the sense of `owl:sameAs`). This introduces many undesirable logical consequences and may contradict with existing schema knowledge. Such problems are even more likely when reusing existing ontologies that do not provide a clear maintenance and evolution strategy as well as by being too careless with the use of `owl:sameAs` links to external (and fluid) datasets.

The problem lies in the attempt to strengthen the semantics of previously under- or informally specified vocabulary terms used to semantically enable data. This is especially problematic for large datasets from different sources that were created and maintained by different parties. In many cases a retroactive "deep semantification" will be difficult or even impossible if it has not been introduced up front.

There is no simple solution for this issue, and a *rapid adoption* approach will sooner or later always lead to such difficulties, semantic aging being another example [17]. At this stage, i.e., to strengthen the semantics of vocabularies, considerable effort will have to be invested in curating the data by mapping it to more expressive ontologies. Regretfully, provenance information for data may already be missing, so that a curation of the data will not always be feasible. In the end there is a trade-off between rapid adoption and ease of establishing deep semantics capabilities, which has to be considered for each use case and application area.

However, some of the overhead work can be avoided by treading carefully from the start. It helps to reuse existing high-quality ontologies and ontology design paterns, and it is important to have a clear understanding of the formal semantics of the adopted ontology language (e.g., OWL), and its implications, even if the initial plan is to only use simple language constructs. To give another elementary example, novices in conceptual modeling often confuse class

[4] FOAF [16] treats email addresses this way, for example.

hierarchies with partonomies, and may be tempted to use `rdfs:subClassOf` as a part-of relationship. The same is true for the more informally used is-a and instance-of relations. By having a clear grasp of the formal semantics of OWL (and RDFS) vocabulary, such mistakes can be avoided.

Summary

We have presented some key aspects concerning the elevation of semantic technologies for adoption in the sciences. In particular, we discussed central value propositions of semantic technologies and ontologies as well as potential roadblocks related to their adoption. While we are aware that the presented list of topics is incomplete and only outlined here, we hope that it will help to start a discussion on how to clarify the value proposition of the Semantic Web within the sciences, communicate paradigm changes and not technologies, lay out roadmaps for knowledge infrastructures such as NSF's EarthCube, and foster our shared visions without overselling them.

Acknowledgement. The second author acknowledges support by the National Science Foundation under award 1017225 "III: Small: TROn – Tractable Reasoning with Ontologies." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Janowicz, K., Hitzler, P.: The Digital Earth as a knowledge engine. Semantic Web Journal (to appear, 2012), `http://www.semantic-web-journal.net/`
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
3. Jones, C.B., Alani, H., Tudhope, D.: Geographical Information Retrieval with Ontologies of Place. In: Montello, D.R. (ed.) COSIT 2001. LNCS, vol. 2205, pp. 322–335. Springer, Heidelberg (2001)
4. Nedas, K., Egenhofer, M.: Spatial-scene similarity queries. Transactions in GIS 12(6), 661–681 (2008)
5. Janowicz, K., Raubal, M., Kuhn, W.: The semantics of similarity in geographic information retrieval. Journal of Spatial Information Science (2), 29–57 (2011)
6. Li, N., Raskin, R., Goodchild, M., Janowicz, K.: An ontology-driven framework and web portal for spatial decision support. Transactions in GIS 16(3), 313–329 (2012)
7. NASA: A.40 computational modeling algorithms and cyberinfrastructure (December 19, 2011). Technical report, National Aeronautics and Space Administration (NASA) (2012)
8. Kuhn, W.: Geospatial Semantics: Why, of What, and How? In: Spaccapietra, S., Zimányi, E. (eds.) Journal on Data Semantics III. LNCS, vol. 3534, pp. 1–24. Springer, Heidelberg (2005)

9. Lund, G.: Definitions of forest, deforestation, afforestation, and reforestation. [online] gainesville, va: Forest information services. Technical report (2012), available from the world wide web: http://home.comcast.net/~gyde/DEFpaper.htm
10. Manola, F., Miller, E.: RDF primer, W3C Recommendation. Technical report, W3C, February 10 (2004)
11. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 305–320. Springer, Heidelberg (2010)
12. Hitzler, P., van Harmelen, F.: A reasonable Semantic Web. Semantic Web 1(1-2), 39–44 (2010)
13. Jain, P., Hitzler, P., Yeh, P.Z., Verma, K., Sheth, A.P.: Linked Data is Merely More Data. In: AAAI Spring Symposium Linked Data Meets Artificial Intelligence, pp. 82–86. AAAI Press (2010)
14. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation, October 27 (2009), http://www.w3.org/TR/owl2-primer/
15. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers (2011)
16. Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.98. Namespace Document, August 9 (2010), http://xmlns.com/foaf/spec/
17. Schlieder, C.: Digital heritage: Semantic challenges of long-term preservation. Semantic Web 1(1-2), 143–147 (2010)

Towards a Spatio-temporal Form of Entropy

Christophe Claramunt

Naval Academy Research Institute,
Lanvéoc-Poulmic, BP 600, 29240 Brest Naval, France
christophe.claramunt@ecole-navale.fr

Abstract. Although information theory is primarily concerned with the transmission of information it can be also applied to the quantification of the intrinsic information that emerges from a given physical system. Over the past years, principles of information theory have been applied to many environmental and ecological studies. However, it still appears that the initial concept of entropy as identified by Shannon's initial contribution cannot be directly applied to evolving geographical systems. The research introduced in this paper suggests an extension of the concept of entropy to the spatial and temporal dimensions, by taking into account the distribution of entities in space and time. We propose a series of entropy measures that together form a set of complementary indices to evaluate the distribution of entities, events and categories in space and time. The whole approach is exemplified by several illustrative configurations.

Keywords: Information theory, entropy, spatio-temporal entropy.

1 Introduction

Things distributed in space hold and embed some intrinsic properties that can be recognised by humans perceiving their environment. Information is conveyed from the world to our minds and some degrees of complexity and order emerge from an observation. Entities can be recognised and generate a sense of legibility as it is frequently the case in cities [11], as well as they afford different activities as asserted by the theory of affordance [5]. Those entities are remembered, and enfold some intrinsic information. The transmission of information has been formally and computationally studied by the theory of information introduced by Shannon [19] and Shannon and Weaver [20]. The main objective of information theory is to evaluate and quantify the information transmitted by a channel. In general, the most specific and diverse the system observed, the more information it is generated and transmitted. The concept of quantifiable information has been formalised by the measure of entropy that evaluates the degree of diversity of the distribution of a variable. Since this seminal contribution, a large amount of work has been developed to apply or enrich the notion of entropy, this denoting a large interest in the application of this concept and the range of domains where it can be useful. When considering small- and large-scale geographical systems, the measure of entropy has been applied by many environmental, ecological (e.g., [8] for a

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 221–230, 2012.

summary) and cartographical studies [18], but still with a lack of complete integration of the spatial and temporal dimensions. In fact the temporal dimension is hardly integrated, and the spatial dimension limited to some local properties such as connectivity, neighbouring and fragmentation relationships [6].

The research presented in this paper revisits the notion of entropy when applied to a system embedded in space and time. We propose to extend the notion of entropy first to space, then to time, and then to space and time. The idea behind is that the entropy of a set of entities distributed in space and time should not only reflect the distribution of those entities over a set of categories, but also their distribution in space and time. Our intention is not to provide a demonstrable extension of the entropy to space and time, but rather to propose a series of indices that reflect and reveal the role those dimensions can play. Another assumption of our work is that such an extension should reflect some of our intuitions when perceiving the structural properties that emerge from a set of entities located in space and time. The work presented is grounded on a previous work where the measure of entropy has been extended to space by a measure of spatial entropy [2], [9] and extended in the present paper to the temporal and spatio-temporal dimensions. The remainder of the paper is structured as follows. Section 2 briefly introduces some basics of the theory of information and its application to environmental systems. Section 3 presents the notion of spatial entropy while section 4 introduces the measures of temporal and spatial-temporal entropies. Finally, section 5 concludes the paper and outlines some further work.

2 Modelling Background

The quantification of the information generated by a given system is a long standing problem in information theory. Information is denoted here as an intrinsic property of a physical system, and the entities present in that system, not as the extrinsic meaning generated. In this way, information theory has been formally studied and represented by the notion of entropy to evaluate how much information is conveyed by a given system [19]. Shannon's entropy is a mathematical indice that measures diversity in categorical data. Indeed, the way entities are grouped in categories in a given system is fundamental to the measure of entropy, those categories being the result of the differences that can be identified between those entities [3], [4]. When applied to the theory of information, the entropy evaluates the degree of choice in the selection of a statement or an entity in a given system, it is more formally given by

$$H = -K \sum_{i=1}^{n} pi \log_2 (pi) \tag{1}$$

where pi is defined as the proportion of the total number N_i of entities of the class i over the total number N of entities, that is, $p_i = \frac{N_i}{N}$, K is a positive constant.

The entropy H is a positive value as the terms $pi \log_2 (pi)$ are negative, and bounded by the unit interval when K is equal to the unit value. For a given number of categories, the entropy is maximum when each class is present in equal proportions, while for an eveness of distribution the entropy increases with the number of classes.

Since the seminal contribution of Shannon the measure of entropy has been widely applied in different ways to ecological and environmental studies. This does not reflect a lack of agreement on a common concept, but rather the fact that the range of application of these measures is large and thus should be adapted to the specific properties of the system represented [8]. Actually, each diversity index encompasses some mathematical properties that exhibit specific behaviors. Often quoted as the notion of diversity, the concept of entropy has been used to analyse the physiognomy of a landscape and the influence of spatial configurations on ecological functionality and biological diversity [12], [15], [13], [7], [21]. For instance, the dominance index measures the extent to which one or a few category types dominate the landscape in terms of class distribution [16]. Fragmentation and spatial heterogeneity indices evaluate the distribution of the number of patches per category, given a region of space [14]. Shannon's measure of diversity has been also extended using an integration of adjacency relationships of first order, as primitive relationships amongst the regions or local cells that compose a spatial system (regions for discrete representations of space, cells for continuous representations of space). These indices evaluate the relative degrees of interspersion, juxtaposition and contagion amongst several classes of region or local cells. For discrete representations of space the measure of adjacency evaluates to which extent regions of a given class are adjacent to regions of another class. The measure of contagion gives the degree to which patches of the same attribute class are clustered [16], [10], it is correlated with indices of attribute diversity and dominance [17]. Contagion can be applied at either the local cell (interspersion and juxtaposition metrics) or patch, that is, region level (contagion metrics). The contagion evaluates to which extent a given landscape is aggregated (i.e., higher values) or dispersed (i.e., lower values). Contagion is inversely correlated to diversity. For a given number of classes, the contagion is minimum when all classes are evenly distributed and equally adjacent to each others. The contagion index is per definition dependent on the adjacency relation, which is a relatively local spatial relations. This leads to a lack of consideration of the overall structure and arrangement of the system studied, distances and relative dispersion of the population. Even in some cases the adjacency relation might be not relevant or non applicable. In fact, most of these limitations come from the fact that these measures were initially applied to continuous representations of space where the objective was to analyse local variance of pixel distributions. In an early work a derivation of a continuous measure of entropy has been applied to the study of a probability distribution over a progressive distance from a given location [1], but still the relative spatial distribution of the categories is not taken into account.

3 Spatial Entropy

In a related work, we introduced a concept of spatial entropy to take into account the role of space when applying a measure of entropy [2], [9]. The idea behind this notion is to consider the primal role of distance in the observation of a given system. This observation is directly inspired by the First Law of Geography that states that "Everything is related to everything else, but near things are more related than distant things" [22]. Taking the argument further, the entropy should augment when distance between different entities decreases, as well as the entropy should augment when the distance between similar entities increases. In order to more formally evaluate those statements we introduced the notion of *Intra-Distance* d_j^{int} of a given class j that evaluates the average distance between pairs of entities of a given class. A second measure, called the *Extra-Distance* $d_j^{ext}$of a given class j, calculates the average distance between the entities of a that class j and the entities of the other classes. More formally

$$d_j^{int} = \frac{1}{N_j \times (N_j - 1)} \sum_{\substack{i=1 \\ i \in C_j}}^{N_j} \sum_{\substack{k=1 \\ k \neq i \\ k \in C_j}}^{N_j} d_{i,k} \ if \ N_j > 1, d_j^{int} = \lambda \ otherwise \tag{2}$$

$$d_j^{ext} = \frac{1}{N_j \times (N - N_j)} \sum_{\substack{i=1 \\ i \in C_j}}^{N_j} \sum_{\substack{k=1 \\ k \notin C_j}}^{N-N_j} d_{i,k} \ if \ N_j \neq N, d_j^{ext} = \lambda \ otherwise \tag{3}$$

where C_j denotes the set of entities of a given class j, N_j the cardinality of C_j, $N$the total number of entities, *di,j* the distance between two entities i and j, l being a parameter taken relatively small.

The *Intra-Distance* d_j^{int} and *Extra-Distance* $d_j^{ext}$are normalized in order to generate values bounded by the unit interval. d_j^{int} is normalised by $Max(d_j^{int})$ which denotes the maximum value of d_j^{int} over the classes j of the population N. Similarly, $d_j^{ext}$is normalised by $Max(d_j^{ext})$. We denote d_j^{*int} and d_j^{*ext} the normalised values of $d_j^{int}$and d_j^{ext}, respectively. This quantitative evaluation of the distance between pairs of similar and different entities supports the introduction of a new measure of diversityHs, called *spatial entropy.* The usual coefficient K of Shannon's measure of diversity is replaced by a fraction that denotes the respective influence of the *Intra-Distance* and *Extra-distance*

$$Hs = -\sum_{i=1}^{n} \frac{d_i^{*int}}{d_i^{*ext}} pi \log_2 (pi) \tag{4}$$

The *spatial entropy* Hs is semi bounded by the real positive interval [0,+∞]. For some given intra- and extra-distance values, Hs is maximum when the classes are evenly distributed. For a given distribution of classes, the *spatial entropy* increases when either the *Intra-Distance* augments, or the *Extra-Distance* decreases. Due to the fact that the coefficient K is not a constant any more, the additivity property of Shannon's diversity is not maintained as K is replaced by an expression which is not constant over the different classes.

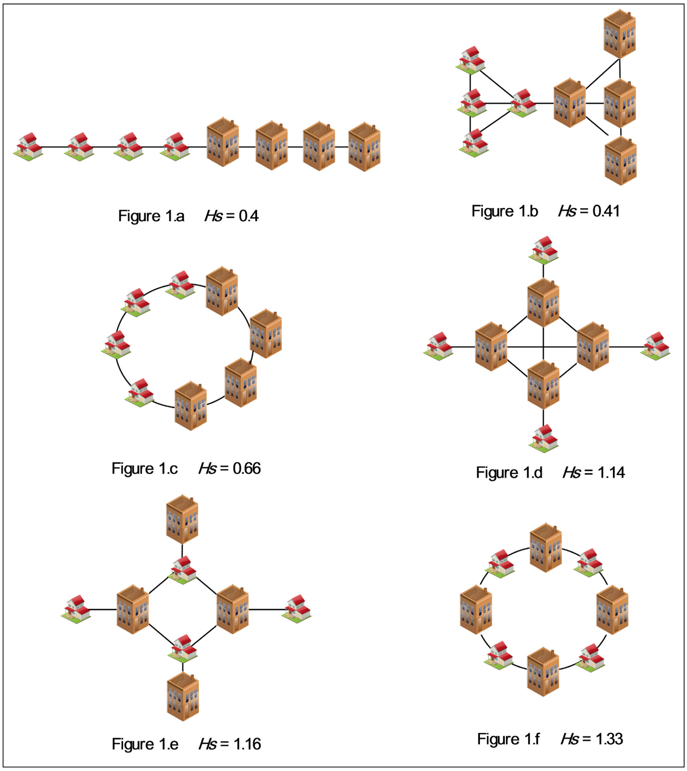


Figure 1.a *Hs* = 0.4

Figure 1.b *Hs* = 0.41

Figure 1.c *Hs* = 0.66

Figure 1.d *Hs* = 1.14

Figure 1.e *Hs* = 1.16

Figure 1.f *Hs* = 1.33

Fig. 1. Spatial entropies

The measure of spatial entropy is illustrated by several primary configurations that emerge from the distribution of houses and buildings in an urban space (figure 1). A direct connection between two of those entities represents an approximated distance of one unit. This schematic example can be considered as a basic example of entities, landmarks or events distributed in space. It appears clearly in figure 1 that the spatial entropy Hs increases progressively when the distances between entities of different classes decrease, and the distances between entities of similar classes increases (and conversely). The spatial entropy is maximum when the two classes of buildings are intertwined in space (figure 1.f). The arrangements exhibited here are local but can be generalized to larger spatial configurations and more categories.

4 Temporal and Spatio-temporal Entropy

When considering a geographical system, entities are located in space, and often in time. This implicitly gives a multi-dimensional component to the concept of distance as this measure can be applied to either space or time. When considering the notion of spatial entropy previously introduced, the role of distance can be now extended to time, by considering a measure of *Intra-TimeDistance* $td_j^{int}$of a given class j that evaluates the average time distance between pairs of entities of a given class. A second measure, so-called the *Extra-TimeDistance* $td_j^{ext}$of a given class j, calculates the average time distance between pairs of entities of a that class j and the entities of the other classes. They are given as follows

$$td_i^{int} = \frac{1}{N_j \times (N_j - 1)} \sum_{\substack{i=1 \\ i \in C_j}}^{N_j} \sum_{\substack{k=1 \\ k \neq i \\ k \in C_j}}^{N_j} td_{i,k} \ if \ N_j > 1, td_j^{int} = \lambda \ otherwise \tag{5}$$

$$td_i^{ext} = \frac{1}{N_j \times (N - N_j)} \sum_{\substack{i=1 \\ i \in C_j}}^{N_j} \sum_{\substack{k=1 \\ k \notin C_j}}^{N-N_j} td_{i,k} \ if \ N_j \neq N, td_j^{ext} = \lambda \ otherwise \tag{6}$$

where $td_{i,k}$ denotes the temporal distance between an entity i and an entity j.

As for the spatial entropy the *Intra-TimeDistance* td_j^{int} is normalised by $Max(td_j^{int})$ which denotes the maximum value of td_j^{int} over the classes j of the population N. Similarly, $td_j^{ext}$is normalised by $Max(td_j^{ext})$. We denote td_j^{*int} and td_j^{*ext} the normalised values of $td_j^{int}$and td_j^{ext}, respectively, the measure of temporal entropy is then given as

$$H_T = -\sum_{i=1}^{n} \frac{td_i^{*int}}{td_i^{*ext}} pi \log_2 (pi) \tag{7}$$

The example illustrated in Figure 2 considers a set of houses and a set of buildings that have been built at some times t_1, t_2, t_3 without consideration of space (in the figure, houses and buildings are materialised at the time of their construction). The time taken into account here is the time of the construction for each house or building. Figure 2.a denotes the case where all houses and buildings have been constructed at a time t_1. Figure 2.b denotes the case where all houses have been constructed at t_1, some buildings at t_2 and others at t_3. Figure 2.c is the case where houses have been built at t_1, buildings at t_3. The temporal entropies that emerge implicitly reflect the degrees of clustering in time. It appears that the temporal entropy progressively decreases when the temporal distances between buildings and houses increase, this reflecting an increase of the degree of clustering in time of those two categories of construction.

When a series of entities and events are located in space and time, their proximity and distribution can be approximated by an evaluation of the distance that relates them, and this by an integration of the spatial and temporal dimensions. Things can be close in space, but far away in time; things can be close in time, but far away in space and so on.

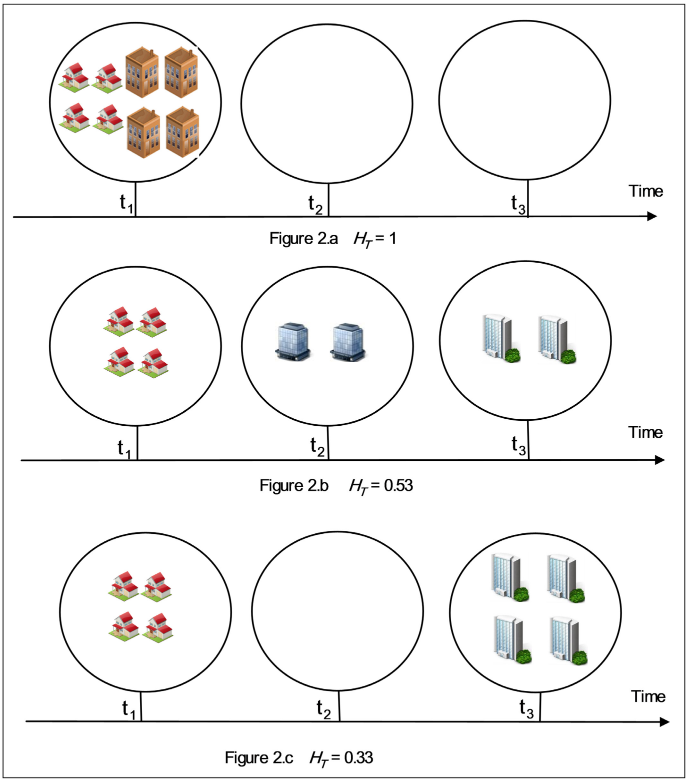


Fig. 2. Temporal entropies

As for the combinations of the two dimensions of space when calculating a distance, one can consider a sort of spatio-temporal distance as a cumulated influence and then as a product of the distances between entities located in space in time. Therefore, spatio-temporal distances can be evaluated as follows

$$std_i^{*int} = td_i^{*int} \times d_i^{*int} \tag{8}$$

$$std_i^{*ext} = td_i^{*ext} \times d_i^{*ext} \tag{9}$$

Then the spatio-temporal entropy can be evaluated as follows

$$H_{ST} = -\sum_{i=1}^{n} \frac{std_i^{*int}}{std_i^{*ext}} pi \log_2 (pi) \tag{10}$$

where std_i^{*int} denotes the average normalised spatio-temporal distance between pairs of entities of the same class, while std_i^{*ext} denotes the average normalised spatio-temporal distance between pairs of entities of different classes.

The example illustrated in Figure 3 summarizes the previous configurations by taking into account space and time, it reflects the typical behaviour of the spatio-temporal entropy. The configurations presented show that the spatio-temporal entropy decreases when buildings and houses are more distant in space (from left

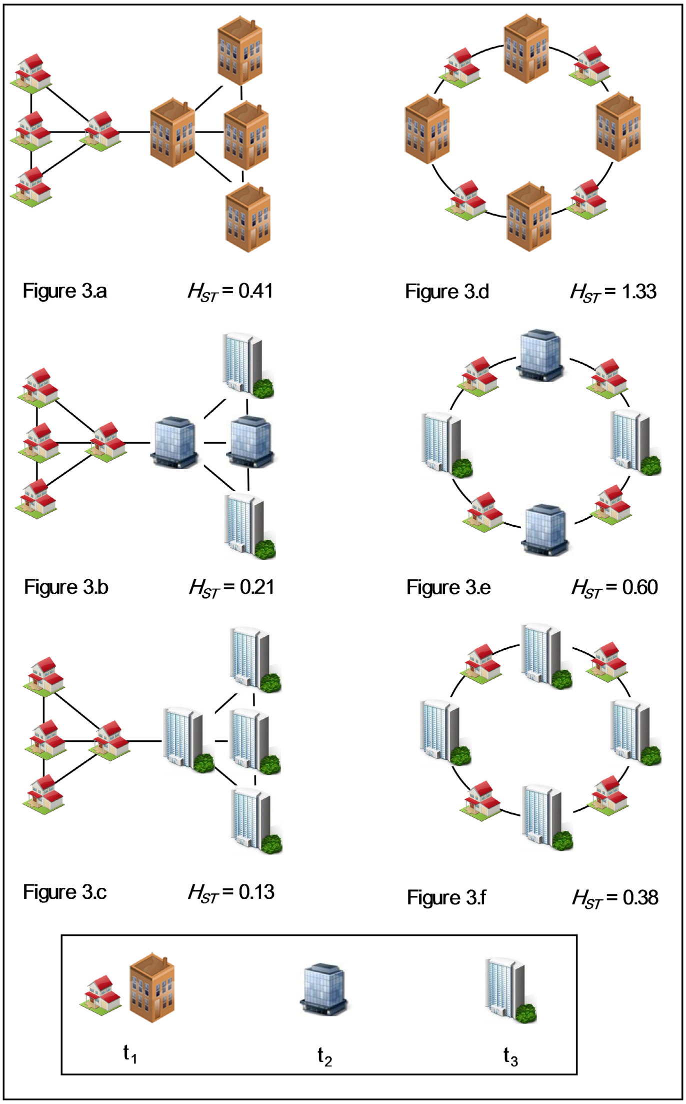


Fig. 3. Spatio-temporal entropies

to right) and time (from top to bottom), this overall reflecting various degrees of clustering in space and time.

When analysed all together, those different entropy values reveal several patterns. They should be considered as a whole or confronted, in order to evaluate some possible trends in space and/or time. It appears that when the roles of space and time concur, the spatio-temporal entropy confirms and increases the values that emerge from the spatial and temporal entropies. When the respective roles of the spatial and temporal entropies differ, the spatio-temporal entropy might evaluate the respective influence of those two dimensions. Indeed the measure of temporal entropy and temporal distance is evaluated by a metrics that should be refined for cyclic phenomena or applications with some specific semantics.

5 Conclusion

The analysis of the distribution of things in space and time is a long standing research issue in many information and environmental sciences. The research presented in this paper proposes an extension of the concept and measure of entropy to the spatial and temporal dimensions. The approach is grounded on the theory of information initially introduced by Shannon. We develop a series of entropy indices that takes into account the role of the distance factor in time and space when evaluating the distribution of categorical data. The measures suggested are flexible as the measures of distance can be computed in different ways. Their formal expression can be also adapted to reflect the semantics of a given phenomena, and also enriched by taking into account additional properties. Those indices can be applied at different levels of abstraction, from large to small scales, and by taking into account different population and categorical data. They provide a specific view of a given distribution of entities in space and time and should be combined with other indices and spatial analysis methods. Also, this approach provides an extension to space and time of the measure of entropy, thus opening several opportunities for a close integration of the theory of information and environmental and geographical studies. Finally, those measures of entropy can support the development of additional reasoning mechanisms, this is an avenue we plan to explore in future work. Ongoing work is oriented to some computational experiments whose objectives will be to proceed performance evaluations.

References

1. Batty, M.: Spatial entropy. Geographical Analysis 6, 1–31 (1974)
2. Claramunt, C.: A Spatial Form of Diversity. In: Cohn, A.G., Mark, D.M. (eds.) COSIT 2005. LNCS, vol. 3693, pp. 218–231. Springer, Heidelberg (2005)
3. Collier, J.: Intrinsic information. In: Hanson, P. (ed.) Information, Language and Cognition: Vancouver Studies in Cognitive Science, pp. 390–409. University of Oxford Press (1990)
4. Deleuze, G.: Differences and Repetitions, p. 350. Columbia University Press (1995)

5. Gibson, J.J.: The Ecological Approach to Visual Perception. Houghton-Mifflin, Boston (1979)
6. Gonzalez, A., Chaneton, E.: Heterotroph species extinction, abundance and biomass dynamics in an experimentally fragmented microecosystem. Journal of Animal Ecology 71, 594–602 (2002)
7. Hurlbert, S.H.: The non concept of species diversity: a critique and alternative parameters. Ecology 52, 577–586 (1971)
8. Jost, L.: Entropy and diversity. OIKOS 113(2), 363–375 (2006)
9. Li, X., Claramunt, C.: A spatial-based decision tree for classification of geographical information. Transactions in GIS 10(3), 451–467 (2006)
10. Li, H., Reynolds, J.F.: A new contagion index to quantify spatial patterns of landscapes. Landscape Ecology 8, 155–162 (1993)
11. Lynch, K.: The Image of the City. MIT Press, Cambridge (1960)
12. Margalef, R.: Information theory in ecology. General Systems 3, 36–71 (1958)
13. McIntosh, R.P.: An index of diversity and the relation of certain concepts to diversity. Ecology 48, 392–404 (1967)
14. McGarigal, K., Marks, B.J.: FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. Gen. Tech. Report PNW-GTR-351, USDA Forest Service, Pacific Northwest Research Station, Portland, OR (1994)
15. Menhinick, E.F.: A comparison of some species individuals diversity indices applied to samples of field insects. Ecology 45, 859–861 (1964)
16. O'Neill, R.V., Krummel, J.R., Gardner, R.H., Sugihara, G., Jackson, B., De Angelis, D.L., Milne, B.T., Turner, M.G., Zygmunt, B., Christensen, S.W., Dale, V.H., Graham, R.L.: Indices of landscape pattern. Landscape Ecology 1, 153–162 (1988)
17. Riitters, K.H., O'Neill, R.V., Wickham, J.D., Jones, B.: A note on contagion indices for landscape analysis. Landscape Ecology 11(4), 197–202 (1996)
18. Pipkin, J.C.: The map as an information channel: ignorance before and after looking at a choropleth map. The Canadian Cartographer 12(1), 80–82 (1975)
19. Shannon, C.E.: A Mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)
20. Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press, Urbana (1949)
21. Tilman, D.: Biodiversity: population versus ecosystem stability. Ecology 77, 350–373 (1996)
22. Tobler, W.R.: A computer model simulating urban growth in the Detroit Region. Economic Geography 46, 234–240 (1970)

The Similarity Jury: Combining Expert Judgements on Geographic Concepts

Andrea Ballatore[1], David C. Wilson[2], and Michela Bertolotto[1]

[1] School of Computer Science and Informatics
University College Dublin, Ireland
{andrea.ballatore,michela.bertolotto}@ucd.ie
[2] Department of Software and Information Systems
University of North Carolina, USA
davils@uncc.edu

Abstract. A cognitively plausible measure of semantic similarity between geographic concepts is valuable across several areas, including geographic information retrieval, data mining, and ontology alignment. Semantic similarity measures are not intrinsically right or wrong, but obtain a certain degree of cognitive plausibility in the context of a given application. A similarity measure can therefore be seen as a domain expert summoned to judge the similarity of a pair of concepts according to her subjective set of beliefs, perceptions, hypotheses, and epistemic biases. Following this analogy, we first define the *similarity jury* as a panel of experts having to reach a decision on the semantic similarity of a set of geographic concepts. Second, we have conducted an evaluation of 8 WordNet-based semantic similarity measures on a subset of OpenStreetMap geographic concepts. This empirical evidence indicates that a jury tends to perform better than individual experts, but the best expert often outperforms the jury. In some cases, the jury obtains higher cognitive plausibility than its best expert.

Keywords: Lexical similarity, Semantic similarity, Geo-semantics, Expert judgement, WordNet.

1 Introduction

Since 2005, the landscape of geo-information has been experiencing rapid and dramatic changes. The concurrent explosion of Web 2.0 and web mapping has resulted in a complex nexus of phenomena, including geo-crowdsourcing, location-based services, and collaborative mapping. Traditional expert-generated geographic information has witnessed the advent of *produsers*, i.e. users engaged in production as well as consumption of spatial data. This resurgence of interest for maps among non-experts online users has been defined Volunteered Geographic Information (VGI) [11]. OpenStreetMap (OSM), a user-generated world map, is a particularly representative instance of these trends.[1]

[1] http://www.openstreetmap.org (acc. June 4, 2012).

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 231–240, 2012.

As diverse communities generate increasingly large geo-datasets, semantics play an essential role to ground the meaning of the spatial objects being defined. In his vision of a Semantic Geospatial Web, Egenhofer stressed that semantic geo-technologies would enable higher interoperability, integration, and effective information retrieval [7]. When dealing with multiple sources of data, common tasks are those of information integration, and ontology alignment. For example, it might be necessary to retrieve the objects representing mountains from two datasets, one labelling them *mountain*, and the other one *peak*. If not supervised by a human, this semantic mapping is very challenging, because of the intrinsic ambiguity and fuzziness of geographic terms.

To identify automatically similar concepts in different datasets or within the same dataset, measures of semantic similarity are needed. Research in semantic similarity has produced a wide variety of approaches, classifiable as knowledge-based (structural similarity is computed in expert-authored ontologies), corpus-based (similarity is extracted from statistical patterns in large text corpora), or hybrid (combining knowledge and corpus-based approaches) [20, 23]. In the area of Geographic Information Science (GIScience), similarity techniques have been tailored on specific formalisms [25, 24, 12].

Typically, geographic concepts in geospatial datasets are described by a short lexical definition. For example, on the OSM Wiki website, the concept of a wetland is described as an 'area subject to inundation by water, or where waterlogged ground may be present.'[2] These definitions are used by data consumers to interpret the meaning of a feature, and by contributors to create appropriate metadata for new features. As the OSM semantic model does not specify fine-grained ontological aspects of the concepts, a suitable approach to compute the semantic similarity of two concepts relies exclusively on their lexical definition. Lexical semantic similarity is an active research area in natural language processing, and several approaches have been proposed [20, 19]. The lexical database WordNet has turned out to be a key resource to develop knowledge-based measures [8].

In general, a judgement on lexical semantic similarity is not simply right or wrong, but rather shows a certain cognitive plausibility, i.e. a correlation with general human behaviour. For this reason, selecting the most appropriate measure for our domain is not trivial, and represents in itself a challenging task. A semantic similarity measure bears resemblance with a human expert being summoned to give her opinion on a complex semantic problem. When facing critical choices in domains such as medicine and economic policy, experts often disagree [17].

Instead of identifying the supposedly 'best' expert in a domain, a possibility is to rely on a jury of experts, extracting a representative average from their diverging opinions [3]. In this study we apply this strategy to the problem of lexical similarity for the domain of OSM geographic concepts, restricting the scope to a set of general-purpose WordNet-based measures. Rather than developing a new measure for geo-semantic similarity, we aim at exploring the idea of combining existing ones into a *similarity jury*.

[2] `http://wiki.openstreetmap.org/wiki/Wetland` (acc. June 4, 2012).

The remainder of this paper is organised as follows. Section 2 reviews relevant related work in the areas of lexical semantic similarity, and WordNet-based similarity measures. The similarity jury is outlined in Section 3, while Section 4 presents and discuss an empirical evaluation. Finally, Section 5 draws conclusions about the jury, and indicates directions for future work.

2 Related Work

The ability to assess similarity between concepts is considered a central characteristic of human beings [25]. Hence, it should not come as a surprise that semantic similarity is widely discussed in areas as diverse as philosophy, psychology, artificial intelligence, linguistics, and cognitive science.

Geographic information science is no exception, and over the past 10 years a scientific literature on similarity for geospatial concepts has been generated [13]. Schwering surveyed and classified semantic similarity techniques for geographic concepts, including network-based, set-theoretical, and geometric approaches [25]. Notably, Rodríguez and Egenhofer have developed the Matching-Distance Similarity Measure (MDSM) by extending Tversky's set-theoretical similarity for geographic concepts [24]. In the area of Semantic Web, SIM-DL is a semantic similarity measure for spatial concepts expressed in description logic (DL) [12].

WordNet is a well-known resource for natural language processing [8]. The usage of WordNet in the context of semantic similarity has fostered the development of numerous knowledge-based approaches, exploiting its deep taxonomic structure for nouns and verbs [15, 23, 16, 26, 1]. Table 1 summarises popular WordNet-based measures [2]. Some measures rely on shortest path between concepts, some include the information content of concepts, and others rely on the WordNet *glosses*, i.e. definition of concepts.

Whilst geospatial measures such as MDSM and SIM-DL can compute context-sensitive similarity in specific ontological formalisms, they cannot be applied directly to the OSM semantic model, in which geo-concepts are loosely described by natural language definitions. By contrast, general-purpose WordNet-based measures are easily applicable to the OSM concept lexical definitions. Spatial-geometric properties of the features – area, shape, topological relations, etc – have a role at the *instance* level, but are beyond the scope of this study, which focuses on abstract geographic *classes*. To the best of our knowledge, WordNet-based measures have not been applied to the geographic domain and, given their high cognitive plausibility in other domains, are worth exploring.

In this paper, we identify an analogy between computable semantic similarity measures and the combination of expert judgements, a problem relevant to several areas. Indeed, expert disagreement is not an exceptional state of affairs, but rather the norm in human activities characterised by uncertainty, complexity, and trade-offs between multiple criteria [17]. As Mumpower and Stewart put it, the "character and fallibilities of the human judgement process itself lead to persistent disagreements even among competent, honest, and disinterested experts" [18, p. 191].

Table 1. WordNet-based similarity measures. *SPath*: shortest path; *Gloss*: lexical definitions (glosses); *InfoC*: information content; *lcs*: least common subsumer.

Name	Authors	Description	SPath	Gloss	InfoC
path	Rada et al. [21]	Edge count in the semantic network	√		
lch	Leacock and Chodorow [15]	Edge count scaled by depth	√		
res	Resnik [23]	Information content of *lcs*	√		√
jcn	Jiang and Conrath [14]	Information content of *lcs* and terms	√		√
lin	Lin [16]	Ratio of information content of *lcs* and terms	√		√
wup	Wu and Palmer [26]	Edge count between *lcs* and terms	√		
lesk	Banerjee and Pedersen [1]	Extended gloss overlap		√	
vector	Patwardhan and Pedersen [19]	Second order co-occurrence vectors		√	

Because of the high uncertainty in complex systems, experts often disagree on risk assessment, infrastructure management, and policy analysis [17, 5]. Mathematical and behavioural models have been devised to elicit judgements from experts for risk analysis, suggesting that simple mathematical methods perform quite well [4]. From a psychological perspective, in cases of high uncertainty and risk (e.g. choosing medical treatments, long term investments, etc), decision makers consult multiple experts, and try to obtain a representative average of divergent expert judgements [3].

To date, we are not aware of studies that address the possibility of combining lexical similarity measures in the context of geographic concepts. This corpus of diverse research areas informs our approach to addressing the problem.

3 The Similarity Jury

A computable measure of semantic similarity can be seen as a human domain expert summoned to rank pairs of concepts, according to her subjective set of beliefs, perceptions, hypotheses, and epistemic biases. When the performance of an expert can be compared against a gold standard, it is a reasonable policy to trust the expert showing the best performance. Unfortunately, such gold standards are difficult to construct and validate, and the choice of most appropriate expert remains highly problematic in many contexts.

To overcome this issue, we propose the analogy of the *similarity jury*, seen as a panel of experts having to reach a decision about a complex case, i.e. ranking the semantic similarity of a set of concepts. In this jury, experts are not human beings, but computable measures of similarity. Formally, the similarity function

sim quantifies the semantic similarity of a pair of geographic concepts c_a and c_b ($sim(c_a, c_b) \in [0, 1]$). Set P contains all concept pairs whose similarity needs to be assessed, while set S contains all the existing semantic similarity measures.

Function sim enables the ranking of a set P of concept pairs, from the most similar (e.g. *mountain* and *peak*) to the least similar (*mountain* and *wetland*). These rankings $rank_{sim}(P)$ are used to assess the cognitive plausibility of sim against the human-generated ranks $rank_{hum}(P)$. The cognitive plausibility of sim is therefore the Spearman's correlation $\rho \in [-1, 1]$ between $rank_{hum}(P)$ and $rank_{sim}(P)$. If ρ is close to 1 or -1, sim is highly plausible, while if ρ is close to 0, sim shows no correlation with human behaviour.

A similarity jury J is defined as a set of lexical similarity measures $J = \{sim_1, sim_2\ \ldots\ sim_n\}$, where all $sim \in S$. For example, considering the 8 measures in Table 1, jury a has 2 members ($J_a = \{jcn, lesk\}$), while jury b has 3 members ($J_b = \{jcn, res, wup\}$).

Several techniques have been discussed to aggregate rankings, using either unsupervised or supervised methods [4]. However, Clemen and Winkler stated that simple mathematical methods, such as the average, tend to perform quite well to combine expert judgements in risk assessment [4]. Thus for this initial exploration, we define the rankings of jury J as the *mean of the rankings* computed by each of its individual measures $sim \in J$. For example, if three measures rank five concept pairs as $\{1, 2, 3, 4, 5\}$, $\{2, 1, 4, 3, 5\}$ and $\{1, 2, 5, 3, 4\}$, the means are $\{1.3, 1.7, 4, 3.3, 4.7\}$, resulting in the new ranking $\{1, 2, 4, 3, 5\}$.

Furthermore, we define ρ_{sim} as the correlation of an individual measure sim (i.e. a jury of size 1), and ρ_J the correlation of the judgement obtained from a jury J. If $\forall sim \in J : \rho_J > \rho_{sim}$, the jury has *succeeded* in giving a more cognitively plausible similarity judgement. On the other hand, when $\exists sim \in J : \rho_J < \rho_{sim}$, the jury has *failed*, being less plausible than its constituent measure sim. A jury J enjoys a *partial success* against sim if $\rho_J > \rho_{sim}$, where $sim \in J$. Similarly, a jury obtains a *total success* if it outperforms all of its members, $\forall sim \in J : \rho_J > \rho_{sim}$.

4 Evaluation

In this section we evaluate the similarity jury, by comparing 154 juries with 8 individual measures, through an experiment on lexical similarity on OSM concepts.

Experiment setup. In order to study the similarity jury, we selected an existing dataset of human-generated similarity rankings on 54 pairs of geographic concepts, collected by Rodríguez and Egenhofer from 72 human subjects [24]. This dataset represents a high-quality sample of human judgements on geospatial similarity, covering large natural entities (e.g. *mountain*, *forest*) and man-made features (e.g. *bridge*, *house*). The concepts of the human-generated dataset were manually mapped onto the corresponding concepts in the OSM, based on their lexical definitions.

Table 2. Results of the evaluation of the lexical similarity on 154 juries. For example, juries of cardinality 2 containing *jcn* obtain a partial success in the 69.3% of the cases.

		Jury containing *sim* (%)								
	$\|J\|$	jcn	lch	lesk	lin	path	res	vector	wup	mean
Partial success $\rho_J > \rho_{sim}$	2	69.3	62.9	84.6	55.0	60.4	79.6	55.0	66.4	66.6
	3	80.1	68.5	84.4	60.5	58.8	86.5	61.8	72.6	71.6
	4	84.4	73.1	83.7	60.4	61.9	87.2	65.6	73.9	73.8
	all	81.3	70.4	84.0	59.8	60.7	86.2	63.2	72.7	72.3
Total success $\forall sim \in J : \rho_J > \rho_{sim}$	2	46.1	42.5	35.7	43.9	42.1	34.6	35.0	42.1	40.2
	3	43.9	37.3	34.9	40.4	31.0	32.0	33.1	36.4	36.1
	4	39.7	32.7	33.9	35.0	28.6	29.5	30.9	33.1	32.9
	all	41.8	35.3	34.4	37.8	30.9	30.9	32.1	35.2	34.8
Plausibility	ρ	.72	.68	.45	.56	.66	.69	.56	.64	.62

To explore the performance of a similarity jury versus individual measures, we have selected a set of 8 *sim* term-to-term WordNet-based measures, $S = \{jcn, lch, lesk, lin, path, res, vector, wup\}$ (see Table 1). The open source project *WordNet::Similarity*[3] implements all of these measures, and was used to compute the similarity scores [20]. As the focus in this study is on the comparison of short segments of text, rather than individual words, the word similarity scores are combined using the technique developed by Corley and Mihalcea [6]. Since the OSM Wiki website holds about 1,900 concept definitions, the complete, symmetric similarity matrix for OSM concepts would contain about 1.8 million rankings.

In the context of risk assessment, large panels with more than 5 experts do not seem to outperform smaller ones [9]. Therefore, we consider the range of jury sizes $|J| \in [2, 4]$ to be appropriate for this study. All the subsets of S of cardinality two, three, and four were computed, resulting respectively in 28, 56, and 70 juries, for a total of 154 juries. The experiment was carried out through the following steps:

1. Compute $rank_{sim}(P)$ for the 8 measures on the OSM definitions.
2. Combine the individual measures into 154 jury $rank_J(P)$ by averaging the $rank_{sim}(P)$ of their members.
3. Compute cognitive plausibility against human-generated rankings for the 8 individual measures (ρ_{sim}) and the 154 juries (ρ_J).
4. Compute partial and total success ratio for juries containing a given *sim*.

Experiment results. Table 2 summarises the results of this experiment, showing the success and total success ratio of the juries containing a given *sim*, and the total success for each measure. The table shows the cognitive plausibility ρ for each measure *sim*, computed against the human rankings. It is possible to note that measures *jcn*, *res*, and *lch* have the highest cognitive plausibility. The jury results are grouped by jury cardinality (2, 3, and 4), and overall results

[3] http://wn-similarity.sourceforge.net (acc. June 4, 2012).

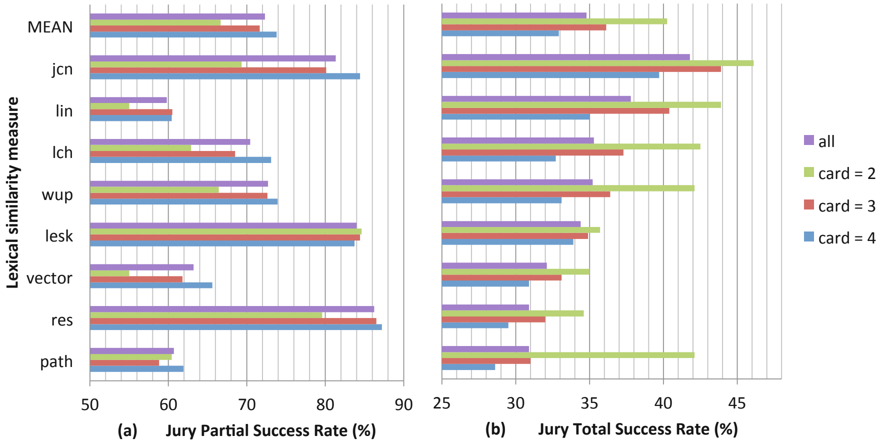


Fig. 1. Results of the lexical jury experiment: (a) partial success of the jury versus an individual measure; (b) total success of the jury versus all its member measures. *MEAN*: mean of success rates; *card*: cardinality of jury J.

(all). The results of the experiment are also displayed in Figure 1, which shows the success ratio of the juries grouped by their cardinality. For example, 80.1% of all juries of cardinality 3 containing the measure *jcn* are better than *jcn* in isolation. These results show a clear pattern: most juries enjoy a partial success over a given *sim* ($> 59.8\%$), while a minority of the juries obtain total success on all of their members ($< 41.8\%$). It is interesting to note that, in the experimental results, the plausibility of a jury is never inferior to that of all of its members, $\exists sim \in J : \rho_J < \rho_{sim}$.

The jury size has a clear impact on the success rate. Small juries of cardinality 2 tend to have a lower partial success ($mean = 66.6\%$), than those with 3 and 4 members (respectively 71.6% and 73.8%). Therefore larger juries have higher chances to obtain partial success over an individual measure. On the other hand, an opposite trend can be observed in the total success of a jury over all of its member measures. Juries of cardinality 2 tend to have a higher total success rate ($mean = 40.2\%$), compared with larger juries ($mean = 36.1\%$ for cardinality 3, and 32.9% for cardinality 4). As larger juries include more measures, it is more likely that one member outperforms the jury.

This empirical evidence shows that in 93.2% of the cases, the jury performs better than the average of the cognitive plausibility of its members, which would be by definition always lower than the plausibility of the best member: if the jury were simply returning the mean plausibility, its total success rate would always be 0%. By averaging the rankings, the jury reduces the weight of individual bias, converging towards a shared judgement. Such shared judgement is not necessarily the best fit in absolute terms, but tends to be more reliable than most individual judgements.

Given that we are measuring the cognitive plausibility of these similarity measures by the correlation with human rankings, the relationship between ρ of *sim* and the jury success ratio needs to be discussed. Interestingly, the cognitive plausibility ρ_{sim} shows no correlation with the jury partial and total success ratios (Spearman's $\rho \approx .1$). This suggests that even measures with high plausibility (such as *jcn* and *res*) still benefit from being combined with other measures. For example, the most plausible measure is *jcn* ($\rho = .72$), so it would be reasonable to expect a low success ratio, given that the measure is the most qualified expert in the panel. This expectation is not met: *jcn* shows a high partial and total success ratio (respectively 81.3% and 41.8%). The juries not only outperform individual measures in most cases, but can also obtain higher cognitive plausibility than its best member.

5 Conclusions and Future Work

In this paper we have proposed the analogy of the *similarity jury*, a combination of semantic similarity measures. The idea of jury was then evaluated in the context of lexical similarity for OSM geographic concepts, using 8 WordNet-based semantic similarity measures. Based on empirical results, the following conclusions can be drawn:

- In the context of the lexical similarity of geographic concepts, a similarity jury J is generally more cognitively plausible than its individual measures *sim* (partial success ratio $> 84.6\%$).
- A jury J is generally less cognitively plausible than the best of its members, i.e. $max(\rho_{sim}) > \rho_J$ (total success ratio $< 46.1\%$).
- In a context of limited information in which the optimal measure *sim* is not known, it is reasonable to rely on a jury J rather than on an arbitrary measure. The jury often outperforms even the most plausible measures.
- The similarity jury is consistent with the fact that, as Cooke and Goossens pointed out, "a group of experts tends to perform better than the average solitary expert, but the best individual in the group often outperforms the group as a whole" [5, p. 644].

In this initial study we have investigated the general behaviour of the similarity jury, by combining term-to-term WordNet-based similarity measures *sim*, in the context of geographic concepts of OSM. Our findings are consistent with those in the area of expert judgement combination for risk assessment [4, 5]. This indicates that the analogy of the jury is sound in the context of semantic similarity measures. However, in order to generalise these results, more work would be needed.

We have adopted a simple technique to combine rankings, i.e. a simple mean. More sophisticated techniques to combine rankings could be explored [22]. Furthermore, the empirical evidence presented in this paper was collected in a specific context, i.e. the lexical similarity of the geographic concepts defined in OSM.

General-purpose similarity datasets, such as that by Finkelstein et al. [10], could be used to conduct experiments across other semantic domains.

The importance of semantic similarity measures in information retrieval, natural language processing, and data mining can hardly be underestimated [25]. A scientific contribution can be given not only by devising new similarity measures, but also by identifying effective ways to combine existing measures. In this sense, we believe that the similarity jury represents a promising direction worth investigating further, given its potential to enhance the cognitive plausibility of computational measures of semantic similarity.

Acknowledgements. Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

References

[1] Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)

[2] Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics 32(1), 13–47 (2006)

[3] Budescu, D., Rantilla, A.: Confidence in aggregation of expert opinions. Acta Psychologica 104(3), 371–398 (2000)

[4] Clemen, R., Winkler, R.: Combining probability distributions from experts in risk analysis. Risk Analysis 19(2), 187–203 (1999)

[5] Cooke, R., Goossens, L.: Expert judgement elicitation for risk assessments of critical infrastructures. Journal of Risk Research 7(6), 643–656 (2004)

[6] Corley, C., Mihalcea, R.: Measuring the semantic similarity of texts. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pp. 13–18. Association for Computational Linguistics (2005)

[7] Egenhofer, M.: Toward the Semantic Geospatial Web. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, pp. 1–4. ACM (2002)

[8] Fellbaum, C.: WordNet: An electronic lexical database. The MIT press (1998)

[9] Ferrell, W.: Combining individual judgments. In: Behavioral Decision Making, pp. 111–145 (1985)

[10] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing Search in Context: The Concept Revisited. ACM Transactions on Information Systems 20(1), 116–131 (2002)

[11] Goodchild, M.: Citizens as Sensors: the world of volunteered geography. Geo Journal 69(4), 211–221 (2007)

[12] Janowicz, K., Keßler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., Bäumer, B.: Algorithm, Implementation and Application of the SIM-DL Similarity Server. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) GeoS 2007. LNCS, vol. 4853, pp. 128–145. Springer, Heidelberg (2007)

[13] Janowicz, K., Raubal, M., Kuhn, W.: The semantics of similarity in geographic information retrieval. Journal of Spatial Information Science (2), 29–57 (2011)

[14] Jiang, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: Proceedings of International Conference on Research in Computational Linguistics (ROCLING X), vol. 1, pp. 19–33 (1997)

[15] Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. WordNet: An Electronic Lexical Database 49(2), 265–283 (1998)

[16] Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, San Francisco, vol. 1, pp. 296–304 (1998)

[17] Morgan, M., Henrion, M.: Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis. Cambridge University Press (1992)

[18] Mumpower, J., Stewart, T.: Expert judgement and expert disagreement. Thinking & Reasoning 2(2-3), 191–212 (1996)

[19] Patwardhan, S., Pedersen, T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: Proceedings of the EACL 2006 Workshop 'Making Sense of Sense' – Bringing Computational Linguistics and Psycholinguistics Together, vol. 1501, pp. 1–8 (2006)

[20] Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::Similarity: measuring the relatedness of concepts. Demonstration Papers at HLT-NAACL 2004 on XX, pp. 38–41. Association for Computational Linguistic (2004)

[21] Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics 19(1), 17–30 (1989)

[22] Renda, M., Straccia, U.: Web metasearch: rank vs. score based rank aggregation methods. In: Proceedings of the 2003 ACM Symposium on Applied Computing, pp. 841–846. ACM (2003)

[23] Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 1, pp. 448–453. Morgan Kaufmann (1995)

[24] Rodríguez, M., Egenhofer, M.: Comparing Geospatial Entity Classes: An Asymmetric and Context-Dependent Similarity Measure. International Journal of Geographical Information Science 18(3), 229–256 (2004)

[25] Schwering, A.: Approaches to Semantic Similarity Measurement for Geo-Spatial Data: A Survey. Transactions in GIS 12(1), 5–29 (2008)

[26] Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138. Association for Computational Linguistics (1994)

Enriching SQWRL Queries in Support of Geospatial Data Retrieval from Multiple and Complementary Sources

Mohamed Bakillah[1,2,*], Mir Abolfazl Mostafavi[1], and Steve H.L. Liang[2]

[1] Centre de recherche en géomatique, Université Laval,
Québec, Canada
[2] Department of Geomatics Engineering, University of Calgary, Alberta, Canada
mohamed.bakillah.1@ulaval.ca

Abstract. Finding relevant geospatial information is increasingly critical because of the growing volume of geospatial data available through distributed environments. It is also a challenge for the ongoing development of the Geospatial Semantic Web. Data brokers provide searchable repositories through which users can generally retrieve the requested data. But generally, these mechanisms lack the capacity to find and combine complementary data from multiple sources. However, such capacity is needed to answer complex queries. In this paper, we propose a new type of system that supports geospatial data retrieval from multiple and complementary sources. The system uses the Semantic Query-Enhanced Web Rule Language (SQWRL) to support reasoning with complex queries and to enable combination of complementary and heterogeneous data. We have developed and included in this system a query expansion method for the resolution of semantic heterogeneities. The proposed system is illustrated with an application example.

Keywords: Geospatial Data Retrieval, Geospatial Semantic Web, Semantic Query-Enhanced Web Rule Language (SQWRL).

1 Introduction

Sound decision-making in the geographical domain involves answering to complex queries, which requires inferring facts from data retrieved from multiple distributed and heterogeneous geospatial data sources. For example, in the field of disaster management, we need data produced by different organizations, at different levels of government and across different regions, on the capacity of emergency services, road networks, demographics, level of risk, etc [Klien et al. 2006]. Several data discovery and retrieval systems, such as catalogue services in Spatial Data Infrastructures (SDIs), are not adapted to answer complex queries, as they do not take into account cases where data coming from multiple and complementary sources must be combined

[*] Corresponding author.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 241–250, 2012.

[Lutz and Kolas 2007]. These systems are not designed to identify complementary data sets, i.e., data sets which can be combined to infer implicit facts. For example, data on wind speed can be used in combination with data on concentration of air particles to assess the level of pollution. These data can come from different sources, in this case, sensor networks. This is an important limitation of retrieval systems in a context where users are expecting to be empowered with tools that can fully take advantage of the large volume of data.

In this paper, we propose a new approach to support the retrieval of heterogeneous data from multiple and complementary sources. We employ an information broker that uses the Semantic Query-Enhanced Web Rule Language (SQWRL). This language enables to identify entities that verify conditions specified with SWRL rules, the candidate rule language for the Semantic Web [Horrock et al. 2004]. For example, this language enables to state that "*if, in a given region, the concentration of air particles exceed x, and the temperature exceeds y, then the level of health hazard is high.*" In our approach, we consider that the conditions in the rule can be verified using data provided by different heterogeneous sources. In addition, we propose a framework where semantic annotations and semantic mappings between application ontologies and external resources support the enrichment of queries to improve the ability of the system to retrieve data, at the same time addressing the issue of data being described with heterogeneous application ontologies.

The paper is organized as follows: in the next section, we present related work on geospatial data retrieval. In Section 3, we present the SQWRL language. In Section 4, the architecture of the system is proposed and details are provided on the approach. Section 5 presents an application example. Conclusions and future work are provided in Section 6.

2 Geospatial Data Retrieval

Geospatial data retrieval aims at finding relevant geospatial data sets over distributed and heterogeneous data sources. In this section, we give a brief overview of representative approaches.

Geospatial data retrieval approaches include, on the one hand, approaches which allow users to submit queries using their own vocabulary through a natural language interface. Such an approach has been proposed, for example, by Zhang et al. [2010]. On the other hand, other geospatial data retrieval approaches enable the user to submit queries formulated only with primitives defined in an ontology, i.e., a formal specification of a conceptualization [Gruber 1993]. While natural language approaches allow users to submit more expressive queries than ontology-based approaches, natural language approaches are also restricted by the ambiguities of natural language, which may refrain from retrieving the relevant data sets [Lutz and Klien 2006]. In this paper, since our aim is not to focus on the resolution of ambiguities generated by natural language, we also adopt an ontology-based approach, such as those discussed below.

The Bremen University Semantic Translator for Enhanced Retrieval (BUSTER), proposed by Vögele et al. [2003], is an early example of ontology-based information broker middleware for geospatial data retrieval. This approach is representative of a category of retrieval approaches that have exploited Description Logics (DL) ontologies, such as Janowicz et al. [2007] and Wiegand and Garcia [2007]. DL, which are underlying the Ontology Web Language (OWL), allow representing classes of individuals (entities) and properties. They also support subsumption reasoning, i.e., the automatic identification of sub-class relationships between classes. In the BUSTER approach, each data source's semantics is formalized with a DL ontology. Each ontology is developed using a common vocabulary defined in a global ontology. The user can select the query concept from one of the ontologies or specify a query with necessary conditions (in term of properties and range of properties). The RACER and FaCT reasoning engines are used to retrieve the concepts that are subsumed by the query concept. While the global ontology makes the different ontologies comparable to each other, assuming that local ontologies can be developed from a global ontology is not always feasible in an open and dynamic environment where sources are developed independently. Lutz and Klien [2006] proposed a similar approach for the discovery and the retrieval of geographic information in SDIs. Their approach is also based on semantic annotations of geographic feature types with DL classes. The DL classes are compared with those that compose the user's queries using a DL subsumption reasoning engine. Similarly to the BUSTER system, this approach retrieves only the classes that are subsumed by the classes in the query. This system does not allow expressing complex queries with conditions as in the SQWRL language. Pursing the work of Lutz and Klien [2006], Lutz and Kolas [2007] used the Semantic Web Rule Language (SWRL), a combination of OWL-DL with sublanguages of the Rule Markup Language (RuleML), to answer users' queries over several data sources in SDIs. În this paper, we propose a geospatial data retrieval approach that builds on the foundations established in the latter approach, using the SQWRL query language. While Lutz and Kolas [2007] assumed that the semantics is shared by all requestors and providers (i.e., they use the same application ontology), in our approach, we do not make this assumption and we rather address the issue of heterogeneous ontologies by proposing a query enrichment approach based on a framework of semantic annotations and mappings among various resources.

3 Semantic Query-Enhanced Web Rule Language (SQWRL)

SQWRL is a query language for OWL that is built on the Semantic Web Rule Language (SWRL) [O'Connor and Das 2009]. SWRL is one of the languages considered to become a standard for expressing rules in the Semantic Web [Horrocks et al. 2004]. SWRL expresses Horn-like rules using OWL classes. A SWRL rule expresses a logical implication between an antecedent and a consequent. This means that when the antecedent is true, the consequent is also true. Both the antecedent and the consequent are composed of atoms. Atoms are statements that can have one of the following forms:

- $C(x)$, stating that the individual x is an instance of the OWL class C;
- $P(x, y)$, stating that the individual x is linked to the individual y via property P, if P is a property between individuals;
- $P(x, z)$, stating that the value of datatype property P for individual x is z.

The following is an example of SWRL rule expressing the conditions for triggering a pollution alert for a region R:

$$Region(?R) \land HasAirParticleConcentration(?R, C) \land HasTemperature(?R, T) \rightarrow PollutionAlertRegion(?R)$$

SQWRL considers a SWRL rule antecedent as a pattern specification for the query; the consequent is replaced with a retrieval specification [O'Connor et al. 2009]. The SQWRL: select operator takes as input the variables used in the antecedent, and issues the individuals that respect the conditions expressed in the antecedent, e.g.:

$$Region(?R) \land HasAirParticleConcentration(?R, C) \land HasTemperature(?R, T) \rightarrow sqwrl{:}select(?R)$$

A SQWRL query must be processed against facts stored in a unique knowledge base. However, we consider that the facts needed to answer a query can come from different sources. In the following section, we propose a retrieval system where the facts that are relevant to answer a query are retrieved from the different sources and stored in a temporary knowledge base against which the query is then processed.

4 SQWRL Approach for Retrieval of Complementary Data

Fig. 1 illustrates the architecture of the approach.

The system is designed around the information broker, which is a mediator between the available geospatial data sources and the user who is seeking for complementary data sets. Through the user interface, the user can specify a SQWRL query, which also specifies how the data coming from multiple sources must be combined. The SQWRL query is processed with the Jess Rule Engine [Eriksson 2004]. The matchmaking services produce the semantic mappings necessary to compare the query with the sources' description. This system is built on principles of standard architectures for the retrieval of data or services, such as proposed by Vögele et al. [2003] and Klien et al. [2006]. However, the first contribution of the proposed approach with respect to existing work is to enhance the information broker with SQWRL to support

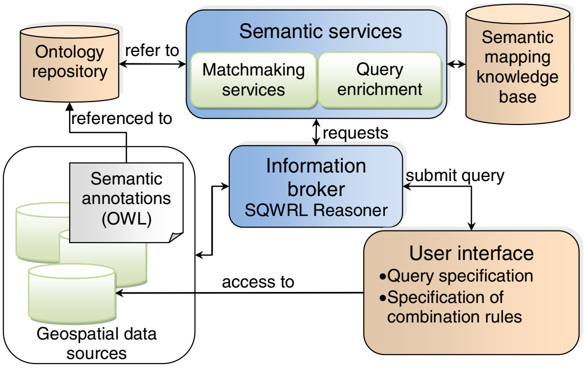


Fig. 1. The architecture

the retrieval of complementary geospatial data sources. In addition, in comparison to existing approaches, we do not assume that all sources are described according to the same application ontology. Although this assumption facilitates retrieval, it is not realistic in the context where available sources describe different application domains. In order to address the issue of heterogeneous ontologies, as a second contribution, we introduce a query enrichment approach. In the following, we introduce the semantic annotations, which support the query enrichment approach, presented in Section 4.2.

4.1 Semantic Annotations

Semantic annotations are defined by Klien [2007] as explicit correspondences (mappings) between the components (classes, attributes, relations, values, etc.) of the data schema of a source and the components (classes, properties, etc.) of an ontology. We also consider that semantic annotations include correspondences between components of an application-specific ontology and components of a more general reference ontology. Semantic annotations enable reasoning with the semantics without altering the local data schemas of sources or application ontology. Semantic annotations can be stored in different ways, i.e., within the source (data source or application ontology), within the target (application or reference ontology), or in a separate source. In this approach, we choose to store semantic annotations in a separate source, as it does not imply altering local sources neither domain or reference ontologies, on which control cannot be assumed. A semantic annotation is formed by a pair of unique identifiers of components from a local source and an application ontology. This association means that the ontology component is the formal representation of the semantics of the local source's component. Because semantic annotations are used to infer which sources contains elements that match a SQWRL query, semantics annotations are formalized with OWL. The establishment of semantic annotations can be a very complex task, difficult to automate, because the names of the data schema elements can include abbreviations or terms known only to the data provider. Therefore, the semantic annotations can be established manually with the help of an ontology editor [Uren et al. 2006]. However, it is out of the scope of this paper to present an approach for establishing semantic annotations.

4.2 Semantic Query Enrichment

The principle of query enrichment is to expand the elements of the query (which are ontology components or values) with other elements that use a different terminology but that have the same meaning. This approach is based on methods for information retrieval described by Boghal et al. [2007] as techniques using "corpus-independent knowledge models," in comparison with approaches that apply knowledge extraction techniques to a set of documents to enrich a query. We assume that the equivalence of meaning is established through a system of semantic annotations and semantic mappings among various resources (Fig. 2). Resources are situated at three levels, i.e., local sources, applications ontologies and global resources. Application ontologies include domain ontologies (describing a knowledge domain, such as ecology, health, etc.)

and task ontologies (designed to support the execution of some activity, such as land use management, disaster planning, etc.). Global resources include reference ontologies, which are domain- and application-independent ontologies, and Linked Data. Linked Data is a Web of data coming from different sources, linked through Resource Description Framework (RDF) predicates [Bizer et al. 2009]. For example, in Linked Data, two entities (e.g., Department of Geomatics Engineering, University of Calgary) can be identified by their Unique Resource Identifiers (URIs) and linked through the predicate "within." As Linked Data contains huge amount of data sets semantically linked to other data sets, it constitutes a rich source to support enrichment of queries. Resources identified in Fig. 2 are linked through semantic annotation and semantic mappings.

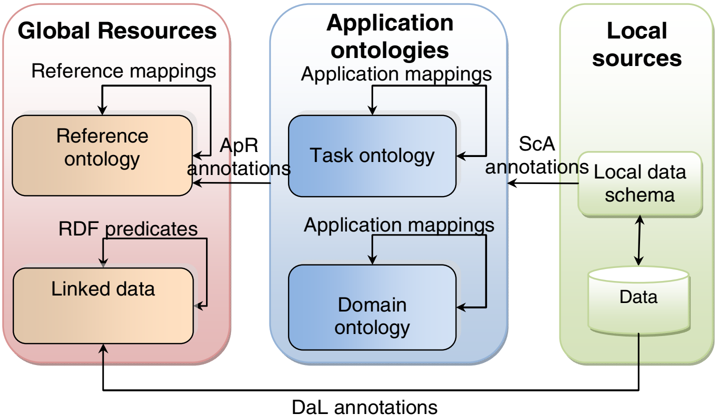


Fig. 2. System of resources, semantic annotations and mappings supporting query enrichment

Semantic mappings link components from the same level, while semantic annotations link components from different levels:

- Components of local sources' data schemas are linked to components of applications ontologies through schema-to-application ontology annotations (ScA annotations, stored in the ScA Annotation Knowledge Base (KB)).
- Components of application ontologies are linked to components of reference ontologies through application-to-reference annotations (ApR annotations, stored in the ApR Annotation Knowledge Base).
- Data from local sources can be linked to URIs on Linked Data through so-called DaL annotations (stored in the DaL Annotation Knowledge Base).

Semantic mappings between ontologies, ScA and ApR annotations support the enrichment of the ontology components that compose queries (classes and properties), while DaL annotations support the enrichment of the values that compose queries. The query enrichment algorithm, provided below, uses mappings and annotations to retrieve elements that can be substituted to components of the query. In this way, a query can be substituted by a set of equivalent queries that use equivalent terms of different ontologies. The enrichment can be horizontal, i.e., a component of a query (which is a component of an application ontology) is replaced with a component of another application ontology, if a semantic mapping that links these components exists. The enrichment is vertical when a component of a query is replaced with a component of a reference ontology, as identified through an ApR annotation. The semantic mappings, which are stored in knowledge bases, can be established manually or through a semantic matchmaking service. For example, in Bakillah and Mostafavi [2010], we have provided a semantic mapping system that can help to support this matching task.

Algorithm 1. Query enrichment

Enrich (query *q*): List <query>

```
Declare and initialize a list of queries equivalent_Query
Add q to equivalent_Query
For all elements el of q
    If el is an ontology component
        Access Application Mapping KB
            For all mappings m where el is a participant
                  Get the relation r stated by m
                  If r == equal
                      Create a copy q’ of q
                  Get el’, the appl. onto. component linked to el through r
                          Replace el with el’ and direct sub-concepts of el’ in q’
                       Add q’ to equivalent_Query
        Access ApR Annotation KB
            For all ApR annotations a where el is a participant
                  Get el’, the reference onto. component linked to el through a
                For all ApR annotations a’ where el’ is a participant
                        Get all appl. onto. components c linked to el’ through a’
                     For all appl. onto. components c linked to el’ through a’
                         Create a copy q’ of q
                         Replace el with c and direct sub-concept of c in q’
                         Add q’ to equivalent_Query
     If el is a value
        Access DaL Annotation KB
        For all DaL annotations a where el is a participant
             Get el’, the name of the Linked Data component linked to el through a
             Create a copy q’ of q
             Replace el with el’ in q’
             Add q’ to equivalent_Query
Return equivalent_Query
```

5 Application Example

Consider a scenario where an employee of a public safety body is responsible for finding an appropriate building where people can be relocated following a disaster or emergency. The employee needs to have access to a data source that contains building in the city, as well as data on the capacity of the rooms that are part of these buildings, in order to assess if the buildings can be used as shelters for at least 100 persons. It is likely that the data needed to find such buildings is not contained in a single source, but that the employee will have to find complementary sources. The employee specifies its information needs through a SQWRL query formulated with terminology defined in a local application ontology:

$$Building(?B) \wedge Near(?B, University\ of\ Calgary) \wedge Room(?R) \wedge PartOf$$
$$(?R, ?B) \wedge HasCapacity(?R, 100) \rightarrow sqwrl{:}select(?B)$$

In this form, it is not likely that submitting this query will enable to retrieve needed geospatial data that uses terminology defined in different application ontologies. Consider that the following ApR annotations have been established between the components used in the query (source components) and the components of reference resources.The reference resources include the specifications of the National Topographic Database of Canada (NTDB), the OpenCyc Spatial Relations ontology, and the WordNet (WN) terminological Database:

Source Component: http://geo-onto.ab.ca/1.0/GeoFeature.owl#Building
Target Component: http://ntdb.gc.ca/ntdb/ManMadeFeature.owl#Building

Source Component: http://geo-onto.ab.ca/1.0/SpatialRel.owl#Near
Target Component: http://sw.opencyc.org/2009/04/07/concept/en/near

Source Component: http://geo-onto.ab.ca/1.0/BuildingFeature.owl#Room
Target Component: http://www.w3.org/2006/03/wn/wn20/instances/word-room

Source Component: http://geo-onto.ab.ca/1.0/SpatialRel.owl#PartOf
Target Component: http://sw.opencyc.org/concept/
Mx4rvVieLpwpEbGdrcN5Y29ycA

Consider a DaL annotation that links the query element "University of Calgary" to the corresponding entry in the Geonames database[1], a Linked Data resource that contains over 8 million toponyms:

Source Component: http://geo-onto.ab.ca/1.0/Place.owl#UniversityOfCalgary
Target Component: http://www.geonames.org/7626260/university-of-calgary.html

Table 1 show how the query is enriched by replacing source components with target components in the query. When target components have sub-concepts, these sub-concepts are used as well to enrich the query. For example, the Building(?B) statement in the query is enriched using the sub-types of http://ntdb.gc.ca/ ntdb/ManMadeFeature.owl#Building in the NTDB specifications. The annotation of places to GeoNames can support resolution of naming heterogeneities for places such as "University of Calgary," which could be also spelled UOfC, Calgary University, etc., in different data sources.

[1] http://www.geonames.org/

Table 1. Example of enrichment of query statements

Original query statement	Enriched query statement excerpt
Building(?B)	BNDT:Building(?B) ∨ BNDT:Arena(?B) ∨... ∨ BNDT:CommunityCentre(?B) ...
Near(?B, UniversityOf-Calgary)	Cyc: Touches(?B, UniversityOfCalgary) ∨ Cyc: AdjacentTo(?B, UniversityOfCalgary) ∨ Cyc: CloseTo(?B, UniversityOfCalgary) ...
Room(?R)	WN: Boardroom(?R) ∨ WN: Hall(?R) ∨ WN: Classroom(?R) ∨ ...
PartOf(?R, ?B)	Cyc: PhysicalPart(?R, ?B) ∨ Cyc: PhysicalPortion(?R, ?B) ∨ Cyc: InternalPart(?R, ?B) ∨ Cyc: ExternalPart(?R, ?B)…
(UniversityOfCalgary)	(http://www.geonames.org/7626260/university-of-calgary.html)

By replacing the components of the query with the target components specified in the annotations and mappings, the query is now expressed using the vocabulary of reference resources. Therefore, semantic annotations that link the target components to vocabulary used in other applications ontologies can be used to rewrite the query using local vocabulary, which will enable to retrieve complementary data stored in sources described with heterogeneous application ontologies.

6 Conclusion and Perspectives

In this paper, we have addressed the issue of retrieving geospatial data from multiple sources. We have pointed out that while there are numerous geospatial data retrieval approaches, very few are designed to retrieve complementary data from multiple sources using a single query; rather, the user has to submit the different queries and perform combination of data, which is a cumbersome task. In this paper, we have proposed a geospatial data retrieval approach that uses the SQWRL language to specify the requested information in a single query. While rule-based approaches to retrieve geospatial data from multiple sources exist, these approaches assume that the same semantics is used to describe the available set of sources. This assumption cannot hold in an environment where sources are provided by different organizations. Therefore, we have coupled the SQWRL approach with a query enrichment approach based on a framework of semantic mappings and annotations between multiple resources. We note that the manual establishment of semantic mappings and annotations is a time-consuming task, since ontologies may be voluminous. While this may be an obstacle, there are numerous semi-automated or automated semantic mapping approaches that can support the user in establishing these mappings (e.g., see Bakillah and Mostafavi [2010]; and the review of Euzenat and Shvaiko [2007]). In future work, we aim to further investigate the role of Linked Data in enrichment of queries. While in this paper, we have shown an example where the GeoNames Linked Data resource can help to enrich queries, further investigations are needed because of the large complexity of resources available on Linked Data.

Ackowledgements. This research was made possible with the funding support from Microsoft Research and Alberta Innovate Technology Future and Natural Sciences and Engineering Research Council of Canada.

References

Bakillah, M., Mostafavi, M.A.: G-Map Semantic Mapping Approach to Improve Semantic Interoperability of Distributed Geospatial Web Services. In: Trujillo, J., Dobbie, G., Kangassalo, H., Hartmann, S., Kirchberg, M., Rossi, M., Reinhartz-Berger, I., Zimányi, E., Frasincar, F. (eds.) ER 2010 Workshops. LNCS, vol. 6413, pp. 12–22. Springer, Heidelberg (2010)

Bhogal, J., Macfarlane, A., Smith, P.: A Review of Ontology-based Query Expansion. Information Processing and Management 43, 866–886 (2007)

Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. Int. Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)

Eriksson, H.: Using JessTab to Integrate Protégé and Jess. IEEE Intelligent Systems 18, 43–50 (2004)

Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)

Gruber, T.R.: A Translation Approach to Portable Ontology Specification. Stanford, California, Knowledge Systems Laboratory Technical Report KSL 92-71 (1993)

Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML (2004), http://www.w3.org/Submission/SWRL

Janowicz, K., Keßler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., Bäumer, B.: Algorithm, Implementation and Application of the SIM-DL Similarity Server. In: Fonseca, F., Rodríguez, M.A., Levashkin, S. (eds.) GeoS 2007. LNCS, vol. 4853, pp. 128–145. Springer, Heidelberg (2007)

Klien, E.: A Rule-Based Strategy for the Semantic Annotation of Geodata. Transactions in GIS 11(3), 437–452 (2007)

Klien, E., Lutz, M., Kuhn, W.: Ontology-based Discovery of Geographic Information Services–An Application in Disaster Management. Computers, Environment and Urban Systems 30, 102–123 (2006)

Lutz, M., Kolas, D.: Rule-based Discovery in Spatial Data Infrastructure. Transactions in GIS 11(3), 317–336 (2007)

Lutz, M., Klien, E.: Ontology-based Retrieval of Geographic Information. International Journal of Geographical Information Science 20(3), 233–260 (2006)

O'Connor, M., Das, A.: SQWRL: A Query Language for OWL. In: Hoekstra, R., Patel-Schneider, P.F. (eds.) Proc. of OWL: Experiences and Directions, Chantilly, Virginia, USA (2009)

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. Web Semantics: Science, Services and Agents on the World Wide Web 4(1), 14–28 (2006)

Vögele, T., Hübner, S., Schuster, G.: BUSTER—An Information Broker for the Semantic Web. KI-Künstliche Intelligenz 3, 31–34 (2003)

Wiegand, N., Garcia, C.: A Task-based Ontology Approach to Automate Geospatial Data Retrieval. Transactions in GIS 11(3), 355–376 (2007)

Zhang, C., Zhao, T., Li, W.: Automatic Search of Geospatial Features for Disaster and Emergency Management. Int. Journal of Applied Earth Observation and Geoinformation 12, 409–418 (2010)

From Polygons and Timestamps to Dynamic Geographic Features: Grounding a Spatio-temporal Geo-ontology

Claudio E.C. Campelo, Brandon Bennett, and Vania Dimitrova

School of Computing, University of Leeds, Leeds, LS2 9JT
{sccec,v.g.dimitrova,b.bennett}@leeds.ac.uk
http://www.comp.leeds.ac.uk

Abstract. This paper presents a knowledge representation approach to modelling and manipulating spatio-temporal data and to grounding a spatio-temporal geographic ontology upon the data. This approach has been developed in the form of a definition-based ontological framework, upon which GIS applications can be developed to perform analysis of geographic phenomena by querying the spatio-temporal database in a more conceptualised fashion. We draw special attention to the representation of geographic features which can change over time, since an appropriate modelling of these dynamic features can provide a natural way of defining other dynamic entities of geographic space, such as events and processes. In addition, the paper discusses some architectural aspects of a GIS which incorporates our semantic model and describes an example of event modelling to illustrate the application of the proposed approach.

Keywords: Spatio-temporal Data Modelling, Geographic Ontologies, Ontology Grounding, Spatio-temporal Reasoning.

1 Introduction

Researchers in Geographic Information Science (GIScience) have investigated means of providing more conceptualised methods of manipulating and querying spatio-temporal data. Recent developments include conceptual models for spatio-temporal data (e.g., [9]), which are frequently described using the entity-relationship model (ER) and Unified Modelling Language (UML). However, despite their expressiveness for describing real-world entities, they lack in providing a method of linking the conceptual and data layers so that reasoning is allowed on spatio-temporal data. Object-oriented approaches have also become of interest (e.g., [12]), since they can provide a model which is both concrete (i.e., implemented in software) and described in a more conceptualised fashion. Nonetheless, inference capabilities of these models are still limited, and consequently queries tend to become more complex and less expressive.

In parallel with this, the scientific community has increasingly realised the value of knowledge representation and reasoning (KRR) approaches to the development of modern GIS. In GIScience, ontologies have been proposed for a

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 251–260, 2012.

variety of purposes; however, the ontology level has been traditionally developed separately from the data level. In this conventional way of designing ontology-based systems, reasoning on queries is performed within the ontology, and data that matches these queries are returned. As a result, the data context can be disconnected from the ontology, which can bring significant limitations to the modelling of dynamic elements of geographic space. We assume that data is a faithful reproduction of physical elements of the world and therefore should be considered to derive coherent descriptions of conceptual entities which are related to these elements.

'*Grounding* gives meaning to ontological primitives by relating them to qualities outside the symbol system, and thus stopping infinite regress' [11, p.01]. Approaches to grounding geographic ontologies have been already proposed. For instance, Bennett et al. [2] presented an approach to grounding vague geographic terms (e.g., river, lake) based on geometric characteristics of water bodies (e.g., linearity, expansiveness). Scheider et al. [11] suggested to ground symbols for qualities (e.g., depth of a lake) by defining them from perceptual/observable primitives (e.g., 'length of a vertically aligned path from the water surface to the bed of a particular water body' [11, p.02]). In the context of this work, we consider that the ontology grounding is established not only when primitive symbols are linked explicitly to elements of data, but also when higher level concepts can be defined in terms of these primitive ones, that is, without concerns about the data structure. For instance, primitive symbols for 'proximity' could be grounded upon a dataset consisting of geographic points (pairs of coordinates) so that higher level concepts, such as 'neighborhood', could be defined without any reference to geographic coordinates.

Considering the temporal dimension (i.e., assuming that qualities of geographic elements are subject to change over time) adds significant challenges to the grounding problem. One might argue that the grounding of temporal information is realised by mapping symbols such as 'instant' or 'interval' to timestamps at the data level. Nonetheless, although this provides a explicit link between the ontology and data levels, we demonstrate that it is not sufficient to make a definite separation between the ontology and the data structure. Methods of grounding geographic ontologies upon the data have been already proposed, however approaches to developing an ontology grounded upon spatio-temporal data seem not to have been sufficiently discussed in the literature, and therefore further developments are required.

This paper presents a KRR approach to representing the spatio-temporal geographic data and to grounding a spatio-temporal geographic ontology upon the data. The discussion given in this paper pays special attention to the representation of *geographic features* which can change over time, as an appropriate modelling of these dynamic features can provide a natural way of defining other dynamic entities of geographic space, such as events and processes. For instance, by understanding the way a forest evolves, one can provide means of identifying events and processes associated with deforestation phenomena. The representation of events and processes is a complex field and is still the subject of substantial disagreements in the literature. Therefore, a discussion on approaches to representing these conceptual entities is beyond the scope of this paper[1]. However, we present an example of event modelling to illustrate the application of the approach proposed here.

The remainder of this paper is structured as follows. The next section overviews some architectural aspects of a GIS which incorporates our semantic model. Following this, Section 3 describes our approach to modelling spatio-temporal data. Then Section 4 presents our approach to representing dynamic geographic features. This is followed by a discussion, in Section 5, on the representation of other dynamic entities in terms of changes affecting geographic features. Finally, Section 6 concludes the paper and points to future directions.

2 Main Architecture

This section describes a typical architecture of a GIS which incorporates our model for representing dynamic geographic entities. This is illustrated in Figure 1. In this architecture, the communication between the *GIS server* and the *data layer* can be established through an *interpretation layer*, which performs logical queries specified in terms of conceptual elements of a *spatio-temporal geographic ontology*. Moreover, the GIS Server can access the spatial-temporal data in the conventional way (i.e., by accessing directly a DBMS or a shapefile), so that map layers generated by these different forms can be overlayed, which is useful to conduct certain analysis. We have built a system prototype to reason about geographic events and processes which adopts this architecture. In this prototype, the components of the interpretation and grounding layers have been developed in SWI-Prolog, whilst the data is stored in a PostgreSQL database.

A *grounding mechanism* provides a way to link explicitly the spatio-temporal geographic ontology and the *spatio-temporal data*, and specific algorithms are applied to ground particular elements of the ontology. Geometric computations required by these algorithms are performed by the *geometry processor*, which contains ad-hoc implementations of geometric operations and also reuse built-in spatial functions provided by a spatial DBMS. The spatio-temporal data may come from heterogeneous sources and therefore may be provided in distinct formats and may have different internal structures. Hence these *raw data* are processed by a designated *data converter*, which converts the data to the format (STAR data model) required by the grounding mechanism.

3 Spatio-temporal Data Representation

Existing spatio-temporal models commonly assume that the *material objects*[2] which inhabit the model are spatially well defined in the data (e.g., a desert

[1] For a comprehensive review of issues and challenges for representing geographic processes, see [4]. For approaches to modelling geographic events and processes see [3,5].

[2] In the context of this paper, such objects are geographic features.

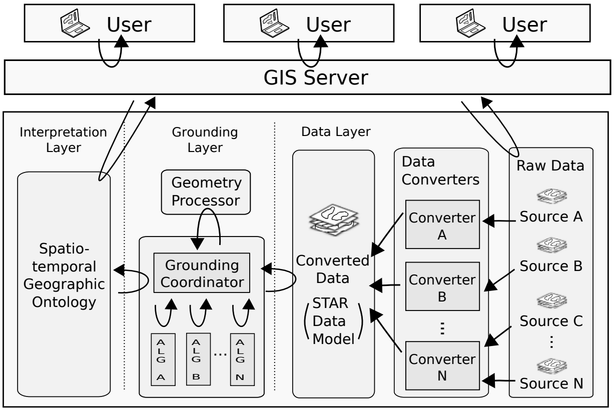


Fig. 1. Typical GIS architecture augmented with our logic-based data model and ontology grounding mechanism

represented as a precise polygon). However, geographic data can be provided in other forms, such as *fields*, which are "measurements on a variable whose value varies through geographic space" [8, p.222]. In this case, as suggested by Galton [7], objects can be inferred from fields (e.g., a desert could be determined from data about precipitation rate).

Our approach to representing the spatio-temporal data aims to provide representational flexibility, so that a wide range of elements can be identified by inferences performed upon a very simple and uniform snapshot-based storage structure. This is also a powerful resource for integration of data originated from distinct sources and at different temporal granularities. This method allows implicit data to be derived and the semantics of time-dependent concepts to be maintained concise thorough continued updates in the database. Furthermore, this approach aims to facilitate the grounding of ontological concepts representing dynamic elements of reality.

The formalism presented in this paper is described in terms of definitions in first-order logic, where free variables are implicitly universally quantified with maximal scope. We employ the Region Connection Calculus (RCC) [10] as the theory of *space*[3]. We assume a total linear reflexive ordering on *time*, and use explicit variables t_i and i_i denoting time instants and intervals, respectively. Time instants variables can be compared by ordering ($\prec$ and $\preceq$) operators and can be quantified over in the usual way ($\forall t[\phi(t)]$). We use the functions $\mathsf{begin}(i)$ and $\mathsf{end}(i)$, which return an instant corresponding to the beginning and the end of an interval i, respectively. The Allen's [1] temporal relations are also employed[4]. We also define the relation $\mathsf{In}(t, i)$ between intervals[5]. The propositional construct $\mathsf{Holds\text{-}On}(\varphi, i)$ asserts that formula φ is true at every time instant t where $\mathsf{In}(t, i)$

[3] The relations *connected* $\mathsf{C}(r_1, r_2)$ and *equals* $\mathsf{EQ}(r_1, r_2)$ are mentioned in this paper.

[4] The relations *partially overlaps* $\mathsf{PO}(i_1, i_2)$ and $\mathsf{Meets}(i_1, i_2)$ are mentioned here.

[5] $\mathsf{In}(i_1, i_2) \equiv_{def} \mathsf{Starts}(i_1, i_2) \vee \mathsf{During}(i_1, i_2) \vee \mathsf{Finishes}(i_1, i_2) \vee \mathsf{Equals}(i_1, i_2)$

holds. The function $\mathsf{ext}(f)$ is also employed, which returns the spatial region corresponding to the spatial extension of a feature f.

3.1 Spatio-temporal Attributed Regions

Our logic-based approach to modelling spatio-temporal data has been named *STAR Data Model* (which stands for Spatio-temporal Attributed Regions). In this model, the spatio-temporal data are stored as triples of the form $\langle a,\ g,\ s\rangle$, which corresponds to the fact that attribute a holds for geometry[6] g at time instant denoted by timestamp s. A broad range of attributes can be associated with geometries. They can be used to describe either types of region coverage[7] (e.g., 'forested', 'arid', 'water covered') or types of geographic features (e.g., 'ocean', 'desert', 'forest'). Polygons denote either spatial regions or spatial extensions of geographic features. Those triples are represented at the logical level as facts of the knowledge base by using the predicate *Spatio-temporal Attributed Region* $\mathsf{Star}(a, g, s)$. The following sortal predicates are also employed to denote four different types of attributes:

- $\mathsf{CAtt\text{-}Hom}(x)$ and $\mathsf{CAtt\text{-}Het}(x)$ are applied, respectively, to denote *homogeneous* and *heterogeneous coverages*. These attributes are associated with spatial regions which are regarded as covered by a single or multiple types of coverages, respectively. Examples of homogeneous coverages are 'forested', 'arid', 'water covered' and 'precipitation $<$ 250mm'. Examples of heterogeneous coverages are 'urbanised' and 'agricultural'.
- $\mathsf{FAtt\text{-}Sim}(x)$ and $\mathsf{FAtt\text{-}Com}(x)$ are applied, respectively, to denote *simple* and *compound geographic features*. While simple features (e.g. desert, road, sea) cannot be composed by other features and every region which is part of them must have the same coverage, compound features (e.g., city, park, beach) may contain other features or spatial regions with different coverages.

The actual denotation employed by these distinct types of attributes depends on the intended application. For example, an attribute named 'forested' can be employed to denote either a homogeneous or a heterogeneous type of coverage. The former might be applied when different types of vegetations are not relevant to the problem at hand, whilst the latter might be employed in association with several homogeneous coverage attributes denoting types of vegetation.

The spatial extension of a geographic feature at a certain time instant can be asserted explicitly or can be inferred as the maximal well-connected region[8] of some particular coverage. *Stars* facts associated with feature attributes can

[6] In the current version of our implementation, geometries are restricted to 2-dimensional simple polygons, which are those whose boundary does not cross itself.

[7] A type of *region coverage* is *not* restricted to types of land coverages. This can also denote *qualities* which can be measured (e.g., by sensors or human observation) and associated with a certain portion of the earth surface, such as 'hot' or 'arid'.

[8] The term 'well-connected region' is used here in agreement with the discussion and definitions given in [6].

be asserted explicitly when the original dataset contains such data. Moreover, certain inferred *Stars* representing spatial extensions of features are asserted explicitly by the system for performance improvement purposes.

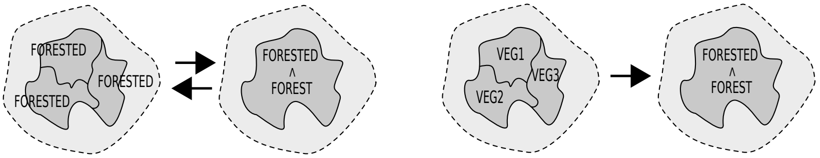
Apart from the facts asserted using the predicate *Star*, other facts can also be asserted using the predicates *Can be Part* $\mathsf{CP}(a_1, a_2)$ and *Must be Part* $\mathsf{MP}(a_1, a_2)$ to determine, respectively, that part-hood relations *can* or *must* hold between *Stars* associated with attributes a_1 and a_2 (e.g. $\mathsf{CP}(\mathit{paved}, \mathit{urbanised})$, $\mathsf{MP}(\mathit{built\text{-}up}, \mathit{urbanised})$). Additionally, facts can be asserted using the predicate *Cannot Intersect* $\mathsf{NI}(a_1, a_2)$, which ensures that spatial regions associated with attributes a_1 and a_2 never overlap (e.g. $\mathsf{NI}(\mathit{urbanised}, \mathit{forested})$). This predicate is useful to support inferences of spatial boundaries between distinct heterogeneous regions as well as inferences of holes affecting geographic features. In addition, a set of axioms is specified to determine inference rules for deriving implicit data and to specify data storage constraints[9].

A formal model $\mathfrak{G}$ of a geographic dataset is $\mathfrak{G} = \langle \mathbb{R}^2, \langle T, \trianglelefteq \rangle, A, \mathcal{D} \rangle$, where: $\mathbb{R}^2$ is the real plane, which represents a portion of the earth's surface under some specified projection[10]. T is the set of all time instants over the time sequence $\langle T, \trianglelefteq \rangle$, where $\trianglelefteq$ is a total linear order over T. A is a set of geographic attributes. $\mathcal{D} \subseteq A \times \mathsf{Poly}(\mathbb{R}^2) \times T$ represents the geographic attributed data as a subset of all possible triples of the form $\langle a,\ g,\ s \rangle$, where $\mathsf{Poly}(\mathbb{R}^2)$ is the set of 2-dimensional simple polygons over $\mathbb{R}^2$. In this paper, we do not intend to present an extensive description of our data representation model, and therefore the full axiomatisation is not given. Rather, we give an informal overview on some key inference rules present in the model. These are as follows.

A. If at a time instant t two spatial regions r_1 and r_2 with the same coverage are spatially connected, then there exists a spatial region which corresponds to their spatial union and has the same coverage of r_1 and r_2 at t.
B. Given two spatial regions r_1 and r_2 with distinct coverages a_1 and a_2, if these regions are spatially connected at a time instant t and there exists a type of coverage a_3 which can comprise the coverage(s) denoted by a_1 and a_2, then the region representing the spatial union of r_1 and r_2 is said to be covered by a_3 at t.
C. Given a spatial region r covered by a at a time instant t, if there exists *no* region r' which at time t contains r and is also covered by a (or covered by a', in case a is a heterogeneous type of coverage and a' is one of the homogeneous coverages which can be present in a), then r denotes the spatial extension of a geographic feature at time t (i.e., a geographic feature is regarded as the maximal well-connected region of some particular coverage).
D. Given a spatial region r covered homogeneously by a, every sub-region of r is also a region with the same coverage of r.

[9] Storage constraints ensure semantic consistency within a dataset (e.g., a heterogeneous region cannot be part of a homogeneous region at a given time instant).

[10] Clearly, one might want to use a different coordinate system or a 2.5D surface model. For simplicity we just assume that the space is modelled by $\mathbb{R}^2$; however, this could easily be changed without modification to the rest of the semantics.



(a) (b)

Fig. 2. Examples of inferences involving attributed spatial regions

E. Given a geographic feature f, there exists a spatial region r with the same spatial extension of f (i.e. $r = \mathsf{ext}(f)$).

Figure 2a illustrates the spatial extension (at a certain time instant) of a simple geographic feature of type ‘forest’ which has been inferred from ‘forested’ spatial regions by applying rule (A) then (C). Note that an inference could have been made in the opposite direction by applying (E) then (D). On the other hand, in Figure 2b, ‘forest’ is regarded as a compound geographic feature and its spatial extension (at a certain time instant) has been inferred by applying (B) then (C). Note that, in this case, the inference could not have been made in the opposite direction.

4 Modelling Geographic Features

We are particularly interested in geographic features which can be modelled as the maximal well-connected regions of some particular coverage. Examples are forests (which can be regarded as the maximum extension of a certain type of vegetation) and deserts (which can be defined based on the level of precipitation). Geographic features are regarded as the material objects which inhabit our spatio-temporal model. They are discrete individuals with well-defined spatial-temporal extensions, are wholly present at any moment of its existence and can change some of their parts while keeping their identity (e.g., a forest can be partially deforested while being still the same forest). Their *identity criteria* is defined in terms of connectivity of their spatial extension over a time interval. We define the operator $f_1 \Bumpeq f_2$, which is true if f_1 and f_2 are geographic features which have the same identity criteria (i.e. f_1 and f_2 are the same individuals).

$$\begin{aligned} f_1 \Bumpeq f_2 \ \equiv_{def} \ & \exists i[\forall i_1 i_2[\,\mathsf{In}(i_1, i) \ \wedge \ \mathsf{In}(i_2, i) \ \wedge \ (\mathsf{PO}(i_1, i_2) \vee \mathsf{Meets}(i_1, i_2)) \ \wedge \\ & \mathsf{Holds\text{-}On}(\mathsf{EQ}(\mathsf{ext}(f_1), r_1), i_1) \ \wedge \ \mathsf{Holds\text{-}On}(\mathsf{EQ}(\mathsf{ext}(f_2), r_2), i_2) \ \rightarrow \ \mathsf{C}(r_1, r_2)\,]] \end{aligned}$$

The *maximum interval* on which a feature maintains its identity is regarded as the interval on which the feature exists (i.e., it is ‘alive’). A *feature’s life* is modelled as a sequence of *Minimum Life Parts* (MLP), which are the shortest stretches of the life-time within which the feature’s spatial extensions are known. In other words, an MLP is a pair of the form $\langle \mathsf{Star}(a, g_1, s_1), \mathsf{Star}(a, g_2, s_2) \rangle$, representing consecutive snapshots of an individual feature. These *Stars* are associated with the same feature attribute a, and can be either asserted explicitly or resulting from inferences performed involving other *Star* (as shown in Figure

2). Figure 3a illustrates the spatial extensions of a feature represented by 7 *Stars*. On the other hand, in Figure 3b, the feature is shown as a spatio-temporal volume, representing an object which occupies a portion of geographic space at any instant of its existence.

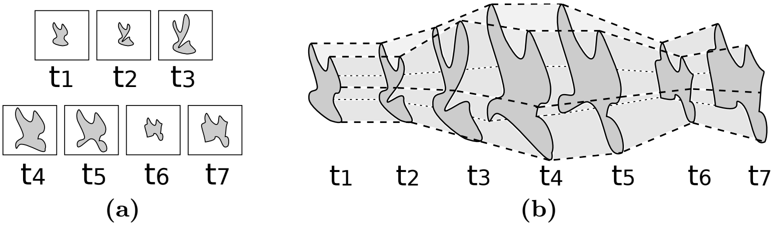


Fig. 3. In (a), the spatial extension of a geographic feature appears in different snapshots; In (b), the feature is shown as a spatio-temporal volume.

Once the concepts *feature*, *feature life*, and *minimum life part* are defined, higher level concepts describing dynamic geographic entities (e.g., events and processes) can be defined in terms of them, that is, without the need to refer to lower level concepts (i.e. *Stars*). This makes the ontology clearly independent from the data structure. The explicit link between the ontology and data levels is established by the definition of an MLP, which is given in terms of *Stars*. The relation $\mathsf{MLP}(f, r_b, t_b, r_e, t_e)$ is specified below, where f, r, t are variables of our logical language denoting, respectively, features, feature types, spatial regions and time instants. This language also includes a set of assignment functions which map variables of the vocabulary to elements of the domain (r_i are assigned to geometries, t_i are assigned to timestamps and u_i are assigned to feature attributes. This relation is defined as follows.

$$\begin{aligned}&\mathsf{MLP}(f, r_b, t_b, r_e, t_e) \equiv_{def} \exists u, r_b, t_b, r_e, t_e[u = \mathsf{feature\text{-}type}(f)\\&\quad \mathsf{Star}(u, r_b, t_b) \wedge \mathsf{Star}(u, r_e, t_e) \wedge t_b \prec t_e \wedge \mathsf{C}(r_b, r_e)] \wedge\\&\quad \neg\exists r', t'[(t_b \prec t' \prec t_e) \wedge \mathsf{C}(r', r_b) \wedge \mathsf{Star}(u, r', t')]\end{aligned}$$

5 Modelling Dynamic Geographic Elements

The approach presented to modelling geographic features can be applied to support the representation of a variety of dynamic geographic elements. To illustrate, we describe an example where an *event* occurrence is defined in terms of spatial changes affecting a geographic feature and how the meaning of conceptual entities can be dynamically adapted to changes in the dataset. First, we define a logical relation *Expands* which compares the area of two spatial regions. This is as follows.

$$\mathsf{Expands}(r_1, r_2) \equiv_{def} \mathsf{area}(r_2) > \mathsf{area}(r_1)$$

Then the following predicate is defined to denote occurrences of expansion events affecting a geographic feature f over a time interval i. For simplicity, we omitted the case where the event occurs in the beginning or in the end of of a feature life.

$$\begin{aligned}&\mathsf{Occurs\text{-}On}(expansion, f, i) \equiv_{def} \exists r_{1b} r_{1e} r_{2b} r_{2e} t_{1b} t_{1e} t_{2b} t_{2e}[\\ &\quad \mathsf{MLP}(f, r_{1b}, t_{1b}, r_{1e}, t_{1e}) \wedge \mathsf{MLP}(f, r_{2b}, t_{2b}, r_{2e}, t_{2e}) \wedge (t_{1e} \prec t_{2b}) \wedge \\ &\quad \neg\mathsf{Expands}(r_{1b}, r_{1e}) \wedge \neg\mathsf{Expands}(r_{2b}, r_{2e}) \wedge \forall r_b r_e t_b t_e[\\ &\quad \mathsf{MLP}(f, r_b, t_b, r_e, t_e) \wedge (t_{1e} \preceq t_b \preceq t_e \preceq t_{2b}) \rightarrow \mathsf{Expands}(r_b, r_e)]]\end{aligned}$$

Observe that the predicate *Star* is not referred to at this level, which makes the definition clearly independent from the data structure. However, a concrete link between the data and logical layers is still maintained, so that changes in data reflects directly the meaning of conceptual elements.

To illustrate how this concrete link is established, suppose a dataset containing 7 *Stars*: $\mathsf{Star}(a, g1, s1)$, $\mathsf{Star}(a, g2, s2)$, ..., $\mathsf{Star}(a, g7, s7)$, where $s_1 \prec s_2 \prec s_3 \prec s_4 \prec s_5 \prec s_6 \prec s_7$ and $\mathsf{area}(g_1) = \mathsf{area}(g_2) < \mathsf{area}(g_3) < \mathsf{area}(g_4) = \mathsf{area}(g_5) > \mathsf{area}(g_6) < \mathsf{area}(g_7)$. Suppose that these elements meet the identity criteria of a feature so that a feature f is inferred, whose life is regarded as composed by 6 MLPs (as shown in Figure 3b). As $\mathsf{area}(g_1) = \mathsf{area}(g_2)$, the area occupied by the feature throughout the first part of its life is said to remain unchanged, and therefore the feature does not expand on this period. On the other hand, the feature is said to expand throughout the second and third MLPs. Given that, the proposition $Occurs(expansion, f, i)$, where $begin(i) = s_2 \wedge end(i) = s_4$, holds. That is, an expansion event is said to occur on the interval comprising the second and third MLPs of the feature f.

Now suppose that some additional data originating from a different source have been integrated into the dataset. These additional data include the element $\mathsf{Star}(a, g_\alpha, s_\alpha)$, which is chronologically positioned between the second and third elements of the original dataset. As a result, a feature f' is inferred (Figure 4), which is said to consist of 7 MLPs rather than 6. Moreover, the period before the event occurrence, on which the feature is said to remain unchanged is now longer, comprising the first 2 MLPs of f. Consequently, the proposition $Occurs(expansion, f', i)$ is false, as the interval on which the feature is said to expand has changed.

From the example given, it can be noticed that, as additional data are integrated into the dataset, an improved (i.e., more detailed) representation of reality is provided at the ontological level. Obviously, the example only illustrates one of numerous ways in which the spatio-temporal dataset may change and the

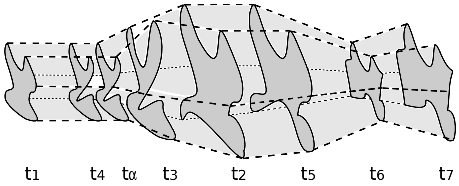


Fig. 4. Geographic feature after inserting a new snapshot into the dataset

variations in the meaning of conceptual entities are dynamically accommodated within the ontology.

6 Conclusion and Further Works

This paper presented a KRR approach to modelling spatio-temporal data and to grounding a spatio-temporal geo-ontology upon the data. It has been shown that modelling the spatio-temporal data in a logical fashion allows us to derive implicit data and provides a natural way to link the data and ontology layers, in order to enable reasoning about dynamic geographic elements. We consider this work as a significant step towards a more concrete integration between conceptualisation and real-world applications in GIS. Further developments include the modelling of additional geometric elements besides polygons, a more complex modelling of a feature life (comprising the modelling of possibles splits and merges), and the extension of the model to a 3-dimensional view of space.

References

1. Allen, J.: Towards a general theory of action and time. Artificial intelligence 23(2), 123–154 (1984)
2. Bennett, B., Mallenby, D., Third, A.: An ontology for grounding vague geographic terms. In: FOIS 2008, pp. 280–293. IOS Press (2008)
3. Campelo, C.E.C., Bennett, B.: Applying standpoint semantics to determine geographical processes instances. In: IOPE, COSIT Workshops (2011)
4. Campelo, C.E.C., Bennett, B.: Geographical processes representation: Issues and challenges. In: Podobnikar, T., Ceh, M. (eds.) Universal Ontology of Geographic Space: Semantic Enrichment for Spatial Data. IGI Global, USA (2012)
5. Campelo, C.E.C., Bennett, B., Dimitrova, V.: Identifying Geographical Processes from Time-Stamped Data. In: Claramunt, C., Levashkin, S., Bertolotto, M. (eds.) GeoS 2011. LNCS, vol. 6631, pp. 70–87. Springer, Heidelberg (2011)
6. Cohn, A.G., Bennett, B., Gooday, J., Gotts, N.: RCC: a calculus for region-based qualitative spatial reasoning. GeoInformatica 1, 275–316 (1997)
7. Galton, A.: A Formal Theory of Objects and Fields. In: Montello, D.R. (ed.) COSIT 2001. LNCS, vol. 2205, pp. 458–473. Springer, Heidelberg (2001)
8. Jacquez, G., Maruca, S., Fortin, M.: From fields to objects: a review of geographic boundary analysis. Journal of Geographical Systems 2(3), 221–241 (2000)
9. Parent, C., Spaccapietra, S., Zimányi, E.: The MADS data model: Concepts to understand the structure of your spatial and temporal data. Journal of Informative Modelling for the Architectural Heritage 0(1), 51–64 (2006)
10. Randell, D.A., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: KR 1992, pp. 165–176 (1992)
11. Scheider, S., Devaraju, A., Janowicz, K., Maue, P., Schade, S., Keßler, C., Ortmann, J., Bishr, M., Fincke, T., Weigel, T., et al.: Grounding geographic information. In: AGILE (2009)
12. Zaki, C., Servières, M., Moreau, G.: Transforming conceptual spatiotemporal model into object model with semantic keeping. In: SECOGIS (2011)

User View of Spatial Networks in Spatial Database Systems

Virupaksha Kanjilal and Markus Schneider*

University of Florida, Gainesville, FL 32611, USA
{vk4,mschneid}@cise.ufl.edu

Abstract. Spatial networks find application in the areas of transportation GIS, network analysis, city planning and others. To effectively use them, it is essential to store spatial networks inside database systems. This paper shows how a user would conceptually view and make use of spatial networks in a database. The discussion has been based on a spatial network data type called *snet* which can store the geometry and the semantics of a network as a single object.

1 Introduction

There is an increasing interest in spatial networks among the researchers in the areas of transportation GIS, network analysis, moving objects databases etc. Road networks, river networks, pipeline networks and other similar networks, which are characterized by a spatial embedding are examples of spatial networks. The increasing interest and use of spatial networks promises the generation of huge amounts of spatial network data which can be stored and handled efficiently only by a database system. Current implementations of spatial networks store the basic components of a spatial network in a database and create an in-memory graph data structure to represent the network. This approach cannot take advantage of the database features like concurrency control or transaction processing. More importantly this approach stores a network by putting it's components into various tables. This makes it impossible to formulate a number of queries. In order to efficiently store spatial network data and perform analysis such as querying, manipulation and other operations on it necessitates the creation of a spatial network data type and incorporating it in available spatial databases. The spatial network data type will enable natively storing spatial networks and running operations on them.

GIS applications demand that spatial database systems, which support GIS applications should store both thematic data as well as geometric data of spatial objects in addition to providing data management operations. So databases should be extended to support spatial network objects. An abstract model of spatial networks built on the concept of infinite point set has been described in

* This work was partially supported by the National Science Foundation under grant number NSF-IIS-0812194.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 261–271, 2012.

[6,5]. This model is a generic one which means it can represent any type of spatial network. By using *labels* to mark thematic attributes, it cannot only store the geometry and topology of the network but also the semantics and the thematic information therein. The data model presents a careful abstract design of a spatial data type and the semantics of related operations. This model serves as a high-level specification for implementation of spatial networks on computers. As computers can only allow discrete representation, the abstract model cannot be directly implemented on computer systems. Based on the abstract specification of spatial networks, this paper shows how a spatial network data type can be elegantly modeled at a logical level by a spatial database type attribute called *snet*. We call this logical model the 'user view' or the 'database view' and the terms convey how a user would conceptually perceive a network object in the database. A single data *snet* object encapsulates an entire spatial network. This conceptual view not only makes the modeling of spatial networks simpler, but also allows various types of queries to be formulated. We demonstrate an SQL-like *Spatial Networks Query Language* (*SNQL*) which may be used to create, access, query, and manipulate a spatial networks within a database system and run operations on it. In order to effectively capture both the spatial, topological configuration, and the semantics of a spatial network, a simple mapping of a network to a mathematical graph is not sufficient. Instead, it is conceptually broken down into simpler components which are finally combined to create a data type termed as *snet* representing a spatial network. *Snet* stores the entire spatial network as a single object so the data for a network need not be spread across several tables and queries can be answered by accessing only a single field in a database table. We specify *snet* as an abstract data type. An abstract data type hides its internal data structure. Retrieval and manipulation of its values can only be performed by high-level operations that conceal how tasks are accomplished. This strategy helps us provide an user interface for the spatial network data structures without specifying the internal structure.

Previous models to implement spatial networks in computer systems has been briefly described in Section 2. The user view of the spatial network in a database along with *Spatial Network Query Language* has been described in Section 3. Future improvements to this model has been discussed in Section 4.

2 Related Work

Graph based approachs to model spatial network [2] simplifies a network to a graph. These approaches completely loses the geometric information in the spatial networks and only maintains the topology of the network. Moreover this solution does not allow the spatial networks to have attributes of their own. Though effective, it has been shown that this method of modeling a spatial network is neither elegant nor robust [9]. [4] provides an elaborate graph-like network data model with three external, user-accessible, representations of the transportation infrastructure, namely the kilometer post, link-node, and geographic representations. Oracle Spatial [7] has a graph-based implementation of a single generic data model for storing spatial networks where the actual network data is separated as links and nodes and are spread across various tables. It has a set of Java API which acts as the middleware layer and provides the functions to access and manage the network data. This kind of data model does not recognize the network as a single entity and hence various parts of the network have to be scattered into a number of tables when stored in a database. This is highly undesirable as common operations (for example, “Do two networks geometrically intersects”) that involve more than one network require the reading of more than one set of database tables. Relational algebra, which is a formal system for manipulating relations or tables in a database, cannot model spatial network queries since it does not have any spatial operators built into it. Moreover, spatial operations cannot be implemented using relational algebra since it is not a full-fledged computer language. Thus it is impossible to formulate SQL queries on them. In order to be useful, these approaches use a *middleware* layer which performs an additional step of interrelating the various representations which means that there needs to be the ability to translate among the different external infrastructure representations. As the middleware layer is outside the database, all the features of the database, for example, transaction processing, concurrency control, querying, etc.are lost. Additionally, the interface they expose and the platforms they work on are very specific, making application development using them difficult. ESRI’s ArcGIS software has specialized network data models [1] for different industries like pipeline, transportation etc. The data is managed by *Geodatabase* [8], which is a middle ware layer like object-relational construct for managing geographic data as features within an industry-standard relational database management system (RDBMS). Another approach taken by some (e.g.[3]) is using routes which correspond to roads or highways in real life and to paths in graphs. They are important conceptual entities as addresses are given relative to routes. Moreover, it is easy to relate network positions to routes.

In order to overcome the disadvantages, our model takes a single entity view of the network, which means all the primitive components and representations of a spatial network are combined into a single data type. This allows a spatial network to be stored as a single entity in a database as a *first class citizen* and all the features of the database can be taken advantage of. Our emphasis in this paper is on an increase of functionality and elegance at the logical level, the propagation of the abstract data type approach in a database context, and the possibility of query support.

3 Database View of Spatial Networks

This section describes how a spatial network data type looks in a spatial database. First we describe what spatial networks are and how they can be modeled in a database using a *snet* data type in Section 3.1. Later we describe a query language called *Spatial Network Query Language* which may be used to create, access and manipulate them in a database (Section 3.2).

3.1 Description of Snet Objects

Spatial networks including transportation networks like road networks for cars, buses, and taxis or railway networks, water pipelines and power networks are a ubiquitous spatial concept. The primary purpose of spatial networks is to provide a spatially constrained environment for materials (in the broadest sense) to move or flow through them. We will first familiarize ourselves with a set of components and features characteristic of all networks. Networks have linear component called *channels* through which flow of material takes place. These are the roads in a highway network or the pipes in a pipeline network. Each channel has a geometry as well as some thematic information attached to it. Examples of thematic information are the name of the road or the speed limit of a road in a highway network. It is assumed that a particular network has a set of attributes associated with it and the set defines the type of the network it is. These channels can intersect among themselves to create *junction points.* Intersection of two roads can be considered as a junction point. Sometimes, the channels cross over or under one another. These points are called *crossover points.* Bridges over a road are examples of crossover points. Since a network can have at a minimum of one channel, a single channel can also be considered a spatial network.

Spatial networks are directly stored in databases as a stand-alone entity as objects of the table attribute type *snet.* An entire *snet* object including the geometry, connectivity and semantics is stored as a single object in the database as a binary large object (blob). The blob has an internal structure which helps quick access and efficient execution of operations on the *snet* object. This mechanism avoids the need for making objects of *snet* data type dependent on multiple tables which in turn makes the data type *snet* a truly first class citizen of the database. The *snet* data type is described as an abstract data type in this paper so as to provide a user's view of spatial networks in databases without specifying how they are implemented. All user access to a spatial network has to be performed using the operators defined in the interface of *snet.* This strategy prevents the user from directly accessing and modifying the internals of a spatial networks object thus potentially rendering it inconsistent.

Snet is a table attribute type, which means we can declare tables having fields of type *snet.* Consider the case of the national power grids in the US, the grids themselves are a type of spatial network and consequently they may be represented as objects of *snet.* The national power grid in the US is categorized into three sectors: Eastern sector, Western sector and Texas sector. These network grids work independently of each other. The actual grids (including the geometry and the thematic data) are stored as objects of *snet* in a table *NatPowerGrids* which has the following schema: *NatPowerGrids(sectorName: String, Grid: snet).* The sector name is stored as a string under *sectorName* and the power grids (the geometry and the thematic data) are stored as *snet* under the attribute 'Grid'. The rows from the table *NatPowerGrids* are of the form:

("Eastern Sector", Grid1)
("Western Sector", Grid2)
("Texas Sector", Grid3)

Even though we consider a spatial network as a single entity, it is conceptually composed of a number of basic components (for example, channels) and in some real life applications it is interesting to access these parts. A set of operators return one or more channels in a network as an object of *snet*. Since the returned objects are of type *snet* too, all the operators defined for *snet* work on them too. The most basic of these operators is *GetChannel*. This operator takes as argument a string which is the id_attr of the channel and returns the channel with the same id_attr. The signature of this operator is $GetChannel : snet \times id\_attr \rightarrow snet$. The *GetAllChannels* operation returns a set of *snet* objects each of which contains a single channel each corresponding to a channel in the original network.

The *snet* data type has a rich set of operations which may be used in *Spatial Network Query Language* introduced in Section 3.2. Below is a small set of operations defined on *snet* data type which has been taken from [5]. These operations take one or more spatial networks as argument as indicated by their signature. The meaning of the operators is intuitively described below.

$$\begin{array}{ll}
Length: & snet \rightarrow real \\
shortestPath: & snet \times point \times point \rightarrow snet \\
getGeometry: & snet \rightarrow line \\
getGeometry: & snet \times id\_attr \rightarrow sline \\
getAttribute: & snet \times string \times point \rightarrow value \\
GetChannel: & snet \times id\_attr \rightarrow snet \\
GetAllChannels: & snet \rightarrow 2^{snet} \\
Union: & snet \times snet \rightarrow snet \\
Intersect: & snet \times snet \rightarrow snet \\
Window: & snet \times region \rightarrow snet \\
Clipping: & snet \times region \rightarrow snet
\end{array}$$

The *Length* operator returns the length of a channel in an snet as a numeric value. There are two versions of *getGeometry*() - the one with only one argument, returns the entire network as a spatial line represented as a *sline*, the second version of this operator takes also an id_attr as argument and returns a spatial line representing the channel which has the same id_attr. Attribute values at a particular point on the network can be obtained by the function *getAttribute*(). This operator takes as arguments the attribute whose value is being sought and the network point at which it should report the value. The operations *Union* and *Intersect* are geometric set operations which take two networks and produce a resulting network which is the geometric union and geometric intersection of the given networks respectively. Given a region and a spatial network, the operations *Window* and *Clipping* extract those parts of the network which intersect with the region. The operation *Window* allows a user to retrieve those *complete* channels of a spatial network whose intersection with a given (region) window is not empty. The *Clipping* operation is similar to the *Window* operation, but it returns only those parts of channels which actually intersect with the specified region.

3.2 Spatial Network Query Language

In this section we show how users can access, manipulate and operate on *snet* object in the database. We first discuss the type of queries possible in a spatial network without assuming that is is a part of a database and then show how SQL like constructs can be extended to provide querying capabilities.

Unlike traditional spatial data objects, which only contain the geometry, spatial networks contain geometry as well as thematic information. Thus a spatial network query may not be purely based on the geometry. Four type of queries have been identified for spatial networks; First, they may be based on the spatial network geometry (*network queries*). This type include queries like "does the bus route 20 intersect with bus route 21". Second, queries may be based on components in a spatial network (*component queries*). These include queries like "How many distinct roads are there in Gainesville" or "Does the 13th Street intersect with the 1st avenue". Third, queries may be based on attributes associated with the spatial network components (*component attribute queries*). These include "What is the capacity of the oil pipe". Lastly, queries may be based on attributes associated with entire spatial network (*network attribute queries*). Example of this type of queries is "Which department administer the highways".

Even though we treat *snet* as an abstract data type object implying that we make no assumptions about how it is implemented, in order to explain the *SNQL*, we lay down some constrains on how the thematic information of a spatial network is stored. The thematic information of any channel in a network is contained in a *label* associated with the channel. These labels are of type *net_label_type* and is essentially a list with each item in the list associated with a *network label attribute.* The network label attribute identifies an attribute in the label. Assuming T = *set of all valid component datatypes for labels*, and A = *set of all possible network label attributes*, formally, a label type is a tuple of pairs : $LS = (a_1 : b_1, a_2 : b_2, \ldots, a_n : b_n)$ where $a_i \in T$ and $b_i \in A$. The first item in any network label is a required attribute called *id_attr* which is a string denoting the name of the channel.

The CREATE LABEL statement is used to create label types in the following manner, CREATE LABEL $L(a_1 : b_1, a_2 : b_2, \ldots, a_n : b_n)$ where $a_i \in T$ and $b_i \in A$. This statement creates a new *net_label_type* L. A label of type L is defined as a tuple $l = (c_1 \in a_1, c_2 \in a_2, \ldots, c_n \in a_n)$. As an example, the label type for the road network containing the speed limits and the number of lanes may be created in the following manner :

CREATE LABEL roadLabel(string: id_attr, real:speed_limit, integer: lanecount)

This statement creates a new *net_label_type* called 'roadLabel' containing three network label attributes: the id_attr of type string,'speedlimit' of type real, and number of 'lanes' of type integer.

In order to create spatial networks of the newly created type *roadLabel* the CREATE SNET NetworkName(net_label_type) statement is used. The statement is written in the following manner :

CREATE SNET road(roadLabel)

The above statement creates an empty network called 'road' which is of type 'roadLabel'. The 'road' network has to be populated with channels and the corresponding label information using the ADD statement. The ADD statement is used to add channels to an already created *snet* object. In order to demonstrate how insertion and querying spatial networks in a database work, we consider a table:

RoadNetworks(RoadType:string, roadNet:snet, AdministeredBy:String)

Before we can discuss the ADD statement, we need the concept of channel chunks. Each channel in a network are conceptually subdivided into elementary parts called *channel-chunks*. *Channel-chunks* are contiguous part of channels where all the points of the channel have identical values for all the thematic attributes. It may be viewed as collecting contiguous points of a channel for which the thematic data are exactly the same. For example, consider a street represented as a channel C_1. This channel is labeled with two attributes *speed_limit* and *laneCount*. Figure 1 shows how these values change over the channel. Thus this channel consist of 4 channel-chunks : s_1(speed_limit = 40, laneCount = 2), s_2(speed_limit = 50, laneCount = 2), s_3(speed_limit = 50, laneCount = 3), and s_4(speed_limit = 40, laneCount = 3).

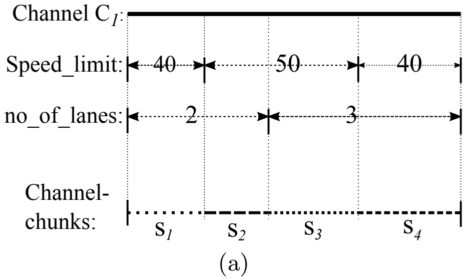


Fig. 1. An Example of a Spatial Network

All these channel chunks have to be inserted into an empty *snet* object individually using the ADD statement. Let us assume that there are two roads in a road network called '13th Street' and 'University Avenue'. Depending on the speed limit, "13th Street" consist of two channel chunks whose geometry is represented by the two *spatial line* objects l_1 and l_2 having speed limits 35 mph and 45 mph respectively. The line objects l_1 and l_2 are internally generated from their textual representation $l1$ and $l2$ respectively by the ADD statement.

Similarly, on "University Ave" the speed limits vary from 35 mph and 45 mph at various portions. These portions are channel chunks and are represented by the two *spatial line* objects l_3 and l_4 (generated from their textual representation $l3$ and $l4$). Both these streets have a lane count of 4. The ADD statement takes a network in which to add the channel chunk, a simple line representing the geometry of the channel, and a label consisting of the label values. In this case the label attributes consist of (id_attr, speed limit, lane count). We use the ADD statements to add these roads in the network in the following manner :

ADD(road, $l1$, ('13th Street', 35, 4))
ADD(road, $l2$, ('13th Street', 45, 4))
ADD(road, $l3$, ('University Ave', 35, 4))
ADD(road, $l4$, ('University Ave', 45, 4))

There can be one or more ADD statement with the same channel name (id_attr) indicating that all the channel chunks are from the same channel. A channel as defined by the abstract model is a continuous linear feature without any breaks or branches. So as to maintain consistency with the abstract definition of channels, the geometry of the chunks of the same channel has to be a continuous sequence. This is known as the *geometric integrity* which states that channels are continuous and cannot have branches. This is ensured by the SNQL's geometric integrity check which ensures that channels are consistent and prevents the user from entering inconsistent data. Whenever a new channel chunk is added using the ADD statement, the SNQL internally checks to see whether chunks of the same id_attr has already been added. If there exists channel chunks with the same id_attr, then the geometric integrity check is performed. The check involves the following : first, the geometry of the newly added chunk should *meet* with the geometry of some other chunk with the same id_attr. *Meet* is a topological predicate which asserts that only the boundary points or the ends of the spatial lines intersect. Secondly, more than two channel chunks of the same id_attr should not *meet* at the same point thus preventing branches in a channel. If any one of the integrity checks fails, the geometric integrity does not hold and the ADD statement results in a failure.

After a spatial network object has been populated, we may insert it in a database table using the regular SQL INSERT statement. Assuming a spatial network called 'interstate_hwy' has been already created, we can insert it into the *RoadNetworks* table in the following manner :

INSERT INTO RoadNetworks VALUES('Interstate', interstate_hwy, 'Federal')

Similarly, all the other records have to be inserted in the table.

Based on the *snet* data type we now define constructs to pose queries in *SNQL*. An SNQL statement has the following clauses: the SELECT clause says what will be returned in the query, the FROM clause indicates a list of tables which may contain some *snet* attribute and the WHERE clause which contains a boolean expression that is evaluated over all the records in a table. Using this scheme, we provide sample queries belonging to the four query types discussed earlier. We assume that the table *RoadNetworks* contains the following tuples:

(“Interstate”, interstate_hwy, ‘Federal’)
(“Country Roads’, cr_road, ‘State’)
(“Limited Access Highway”, laHwys, ‘Federal’)
(“Single Carriage Way”, sig_way, ‘State’)
(“Winter Roads’, win_roads, ‘State’)

The database also has the table *NatPowerGrids* as discussed in Section 3.

Network queries deal with the the entire network as a whole. They include queries like ‘which part of each of the networks intersect with the Interstate’. In formulating this query, we use the spatial network operator *intersect* which returns the intersection between two networks. The SNQL statement for the above query is as follows:

select Intersect(N.roadNet, M.roadNet)
from RoadNetworks **as** N, RoadNetworks **as** M
where M.roadNet = ‘Interstate’

This query performs a self join, then uses the *intersect* operator pairwise with each road network type and the interstate network.

Component queries deal with the components in a network. Channels are the components of a network, and these queries extract part of a channel or complete channel and operate on them. The SNQL can handle these operations since a channel or part of a channel is also an *snet* object. A traditional query in transportation is the shortest path finding which may be of the form ‘find the shortest route from Miami to Atlanta avoiding Alachua county’. This is a restricted form of shortest path since it has to avoid all the roads passing through Alachua county. Assuming that Alachua county is given as a region with the name ‘alachua’, and the initial and the final points as p_A and p_B respectively, we write the query in SNQL as

select ShortestRoute(sn, p_A, p_B)
from RoadNetworks **as** N
where sn **in** Difference(N.roadNet, window(N.roadNet, alachua))

The *window* operation returns the part of the network which intersects with Alachua. This portion of the network is subtracted from the original network (using the *difference* operator) and the shortest path is computed on the truncated network.

Next we provide examples for *component attribute queries*. These queries are based on the thematic information attached to the components of a spatial network. As discussed earlier, each channel of a network has certain attributes attached to it in the form of labels. These values can be extracted by the operator *GetAttribute* which takes a spatial network and a label attribute and returns the value of that attribute corresponding to the network. For example an interesting query is 'find the average capacity in each of the power grids'. This query asks for the average transmission capacity of each of the power lines in each of the grid. We assume that the network label attribute 'capacity' is in the labels for the power grid networks in the table *NatPowerGrids*. We again use the *GetAttribute* operator to find the capacity of each of the power lines in a network and then average the value. This query demonstrates how SQL constructs like GROUP BY may be used in a traditional manner in SNQL to group by the average capacity for each power grid. The SNQL statement is as follows:

```
select    G.sectorName, AVG(GetAttribute(sn, capacity))
from      RoadNetworks as G,  GetAllChannels(G.Grid) as sn
group by  G.sectorName
```

The GROUP BY clause groups the average capacities by each network grid as each network grid has a unique sectorName which the SELECT clause returns.

The final type of queries termed as *network attribute queries* query the attributes attached to an entire network. These attributes are not attached to the *snet* object, but are part of the record containing an *snet* object. For example, 'which government entity maintains the road network' can be answered by a simple SQL query which selects out the 'administered by' field from the *RoadNetworks* table :

```
select N.RoadType, N.administered_by
from   RoadNetworks as N
```

4 Conclusions and Future Work

This paper describes the user view of spatial networks in a spatial database. The single data type *snet* captures the geometry, connectivity and the thematic information in a spatial network and can be used as a table attribute type. Using the newly introduced data type *snet*, spatial networks can be stored in a database as a single entity which prevents it from being spread over a number of tables, thus making a spatial network a first class citizen of the spatial database. In order to access and manipulate the *snet* objects in a database, we described an SQL-like query language called SNQL. SNQL supports the creation, insertion, and manipulation of *snet* data objects in a database table as well as the execution of operations on them.

This paper is part of a complete spatial networks package called *Spatial Networks Algebra* (*SNA*) that can model a broad range of spatial networks and will have a comprehensive collection of operations and predicates defined on them and is aimed at incorporation in the database system.

References

1. http://support.esri.com/en/downloads/datamodel
2. Dale, M.R.T., Fortin, M.-J.: From Graphs to Spatial Graphs. Annual Review of Ecology, Evolution, and Systematics 41(1), 21–38 (2010)

3. Guting, H., de Almeida, T., Ding, Z.: Modeling and Querying Moving Objects in Networks. The VLDB Journal 15(2), 165–190 (2006)
4. Jensen, C., Pedersen, T., Speičys, L., Timko, I.: Data Modeling for Mobile Services in the Real World. In: Hadzilacos, T., Manolopoulos, Y., Roddick, J., Theodoridis, Y. (eds.) SSTD 2003. LNCS, vol. 2750, pp. 1–9. Springer, Heidelberg (2003)
5. Kanjilal, V., Schneider, M.: Modeling and Querying Spatial Networks in Databases. Journal of Multimedia Processing and Technologies 1(3), 142–159 (2010)
6. Kanjilal, V., Schneider, M.: Spatial Network Modeling for Databases. In: ACM Symposium on Applied Computing, ACM (2010)
7. Kothuri, R.V., Godfrind, A., Beinat, E.: Pro Oracle Spatial for Oracle Database 11g (Expert's Voice in Oracle). Apress, Berkely (2007)
8. Zeiler, M.: Modeling Our World: The ESRI Guide to Geodatabase Design. Environmental Systems Research Institute (1999)
9. Miller, H.J., Shaw, S.-L.: Geographic Information Systems for Transportation. Oxford University Press (2001)

A Collaborative User-Centered Approach to Fine-Tune Geospatial Database Design

Joel Grira[1], Yvan Bédard[1], and Tarek Sboui[2]

[1] Department of Geomatic Sciences and Centre for Research in Geomatics,
Université Laval, Quebec, Qc, G1K 7P4, Canada
[2] National Center of Scientific Research, Paris, 75014, France

Abstract. The Geographical Information System (GIS) design process usually involves experts in specific application domains (e.g. geology and forestry). These experts are responsible for capturing user needs, and communicating them to GIS developers. However, as the community of users is not directly involved in the design process, these experts may miss some user intentions. This may lead to ill-defined requirements and ultimately in higher risks of geospatial data misuse. In this paper, we present a collaborative user-centered approach in the design process that aims at improving requirements collection and description through a web 2.0 philosophy of having a more active involvement of users. The approach consists of 1) analyzing the role users could play based on requirement engineering guidelines, and 2) allowing users to iteratively describe their intentions of data usage in given contexts.

Keywords: geospatial database design, collaborative approach, risks of geospatial data misuse, user-centered design.

1 Introduction

For many years, Geographical Information System (GIS) and geospatial data have been designed by teams composed of experts in Geospatial Information Technology - GeoIT (e.g. software engineers, geomatics engineers, database designers) and experts in specific application domains (e.g. geology, forestry, urban planning). Application domain experts (usually one or two) are the representatives of the community of end-users who hold the knowledge of what is needed. Such knowledge includes intentions of use which are expressed in terms of purposes.

Design teams usually assume that the needs of the end-user community are exhaustively collected from their representatives, i.e. the application domain experts who are part of the design teams [33]. However, as these experts usually communicate with only a small number of end-users (or even none at all), some intentions or restrictions of data usage related to specific contexts may remain uncaptured and often assumed. Existing or potential problems with existing data or with the new database design may go unnoticed and lead to ill-defined specifications, undermine the fitness-for-use of the system or lead to inappropriate usages [14].

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 272–283, 2012.

The aim of this paper is to present a collaborative user-centered approach that aims at improving system requirements collection and description as well as reducing the risks of data misuse through a more active involvement of users in the design process.

In the next section, we first review research in the field of requirement engineering (RE). In Section 3, we describe how users' involvement in the design process could help in detecting new requirements and improving existing ones based on RE guidelines. In Section 4, we show how intentions of use can be captured and recorded, in terms of purposes, within a specific context. We conclude and present further work in Section 5. The more specific goal of reducing the risks of data misuse is not treated in this paper.

2 Reviewing Research in Requirement Engineering

Literature about RE confirms that "The success of a software system depends on how well it fits the needs of its users and its environment" (e.g. [3]). Requirements comprise these needs, and RE is the process by which the requirements are determined. Thus, RE is about defining the problem that the software is to solve (i.e. defining "what" the software should do), whereas other software engineering activities are about defining a proposed software solution. RE artefacts (i.e. the output of the RE process) have to be understandable and usable by domain experts and other stakeholders, who may not be knowledgeable about computing.

Literature about RE research has defined five types of requirements tasks: elicitation, modeling, requirements analysis, validation/verification, and requirements management. This decomposition helps to get a high-level overview of research activities.

1 ***Requirements elicitation*** comprises activities that enable the understanding of the purposes for building a system. Most of the research in requirements elicitation focuses on the techniques used to improve the stakeholders' identification [10] and helping them to express their needs [22]. Some other research focus of the RE elicitation techniques used to capture requirements [28].

2 ***Requirements modeling*** consists on expressing requirements in terms of precise and unambiguous models. This process helps to evoke details that were missed in the initial elicitation. The resulting models could communicate requirements to the design team. Modeling notations are the main research focus and differ by the specific details they elicit and record (e.g. data, functions, properties, behavior) [6].

3 ***Requirements analysis*** comprises techniques for evaluating the quality of recorded requirements. Some studies look for errors in requirements [20] or focus on anomalies in requirements [35]. These studies reveal misunderstandings or ambiguity about the requirements that usually call for further elicitation. Risk analysis [26] is part of the requirements analysis techniques that help IT designers to better understand the requirements, their interrelationships, and their consequences.

4 *Requirements validation* ensures that stakeholders' needs are accurately expressed. Thus, validation is typically a subjective evaluation of the requirements with respect to informally described or undocumented requirements. Accordingly, the validation task requires stakeholders to be involved in reviewing the requirements artifacts [19]. Research in this area focuses on improving the information provided to the stakeholder [9].

5 *Requirements management* comprises a number of tasks related to the evolution of requirements over time. It is an umbrella activity. Research in this area includes techniques to ease and possibly automate the identification and documentation of the requirement traceability [25].

As seen above, in non-spatial domain, the RE research community has made significant progress along many fronts. However, not all RE principles are systematically applied in practice [7, 17]. One of these main principles is involving the end-users in requirements definition. In fact, evidences exist about the lack of user involvement in GIS design.

In the next section, we describe how user involvement in the GIS design could help in detecting new requirements, in improving existing ones and in avoiding potential data misuses based on RE guidelines.

3 User Involvement in GIS Design

In spatial database design, there is typically an intended purpose to be considered as this purpose has immediate impacts on the level-of-detail (LoD), precision and topological properties of the geometry representing the spatial objects or phenomena to be displayed on maps. The choice and definition of these geometric properties (i.e. the semantics of the geometry) have immediate impacts on the types of spatial analyses to be performed and on the quality of the results. A lack of adequate understanding of the users' purposes during the design may lead to unsatisfactory data usages. Similarly, if data is used for purposes different from the intended ones, inappropriate results may occur. Several incidents and deadly accidents have been reported in newspapers and Court decisions with regard to inappropriate usages of data, intended or not [33, 23] while the professional liability of not properly warning spatial data users has been raised [11, 12, 34].

Design is not systematically performed with respect to RE guidelines [7]. In fact, the intentions of the user regarding his potential usages of the data are not systematically captured during the requirements elicitation stage: this leads to a *de facto* exclusion of some needs and warnings from the scope of the GIS to be designed.

Moreover, missing requirements constitute the most common type of requirements defect [31]. Referring explicitly to RE techniques might help finding undiscovered requirements and improving existent ones. However, still the role that users should play in each of the RE steps is usually ill-defined, misunderstood or ignored [8, 17, 18] as it is the case in complementary approaches like prototyping and agile design.

In the next section (3.1), we first analyze the roles end-users could play in GIS design. We define how users could be involved in each of the following steps, with regards to the RE tasks and techniques:

1. capturing the purposes of end-users and their representatives
2. understanding the contexts in which the data/system will be used
3. modeling, analyzing, negotiating, and documenting requirements
4. validating that the documented requirements match the negotiated requirements

We then (Section 3.2) propose to fine-tune the GIS design process with a collaborative approach by considering the intentions of use for geospatial data elements in relation to a defined context.

3.1 Analyzing Users' Roles in GIS Design

In this section, we analyze the roles end-users should play in GIS design according the previously listed steps.

A. Capturing the Purposes of Geospatial Data Usage

Elicitation of user intentions and his usage context is a prerequisite for the design process. For database design, this implies one should focus on the data needs rather than on the structure of data. However, traditional design techniques typically start with the analysis of data requirements, which literally constraints the structure of data to be stored, and do not address the usage purposes (i.e. the context) as a main issue. Capturing the usage context needs to be supported by an appropriate RE approach which is purpose-oriented rather than data structure-oriented.

Some progress has been made in the field of user-centered RE [1, 2] but still a gap exists between the "universe of discourse" (UoD)[1] and the produced specifications (see Figure 1). In fact, late discovery of relevant use cases means that at least one usage purpose or a domain concept has been missed. This typically occurs during the different transformations performed throughout the requirement development process (RDP) (Figure 1). Transformations should maintain the link between the UoD and the models without disregarding usage purposes and intentions.

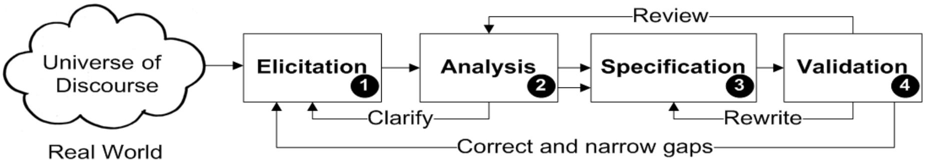


Fig. 1. A Requirements Development Process (RDP) [*17 adapted*]

[1] UoD: view of the world that includes everything of interest [16].

B. Understanding the Contexts

The state-of-the-art techniques for database design assert that design must consider users' possible activities, tasks and intentions [32]. These aspects are part of the intended usage context of the end-user. Accordingly, design must target two different realms: the application domain model and its underlying context.

Usage context modeling (i.e. formalizing the context's relationship with the data) deserve appropriate consideration because context is what defines the scope of the end-user needs [27]. Nevertheless, classical data models (e.g. conceptual models) are more suited to application domain design because they are rooted in traditional design approaches. Few of these approaches have explicitly considered the usage intentions and purposes because their focus has primarily been on products (i.e. systems), operations (i.e. features) and entities (i.e. components) rather than on purposes. We assert that an appropriate paradigm (e.g. goal-oriented) to express the user intentions may help overcoming the limitations of traditional design approaches [24].

C. Analyzing the Usage Contexts

Evidence exists in literature about the human selective cognitive perception: only a subset of interest is extracted among the whole set of the real-world entities [13]. The extracted subset of "domain concepts" of interest is (1) communicated by the application domain experts to the experts in geomatics and in GIS design. Next, it is (2) translated into specifications in order to be implemented.

RE techniques suggest one should start from usage intentions of the end-user (i.e. the usage context model), and derives from them functional and non-functional requirements through a systematic process. However, a number of entities of interest (i.e. part of the UoD) do not reach the specifications. This constitutes a leak that may undermine the transformation from requirements to specifications [5] and lead to risky situations further [33].

D. Validating Requirements

With regards to the requirements validation, the concerns and perspectives of system designers are typically different from those of experts in geomatics and also from those of end-users [33]. Consequently, the validation process is constrained because of inconsistency between the different expectations: in fact, the targeted design is not usually performed against a commonly defined set of expectations. However, the awareness of the necessity to bridge this gap early in the design process has been raised and solutions proposed [34].

3.2 Enriching GIS Design: Collaborative Specification of Context

Based on the user roles analyzed in Section 3.1, we propose an approach that aims to enrich the GIS design process. This approach is iterative, collaborative and purpose-oriented, and it is mainly based on the following two components: (1) the usage context and its structural elements and (2) the process that allows defining it.

(1) Elements of the usage context: purposes, domain concepts and domain concepts mapping

As seen previously (Section 3.1.A), purpose elicitation determines the boundary of the UoD for the model-to-be. In relation to the requirements elicitation task in RE (Section 2.1 and 3.1.A), the elicitation of **purposes** reveals important **domain concepts** for which data need to be collected and stored. These domain concepts and their associations form the data requirements that better capture stakeholders' information needs. The set of these requirements and rules enriches the way of passing from the real world to the data (i.e. specifications) as shown in the Figure 2.

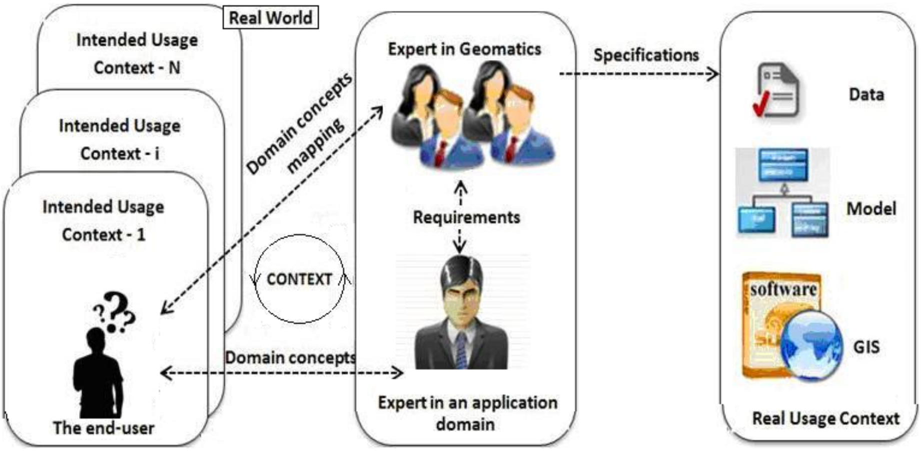


Fig. 2. User roles in the requirements development process

With regards to the missing requirements we outlined above (step 3.1.C), we define in our approach an index-based **mapping between the end-user purposes and geospatial metadata** (e.g. geospatial data quality information) in order to facilitate the requirements derivation process.

As mentioned in the step 3.1.D, the validation step aims to narrow the gap between the documented (i.e. approved) requirements and the expectations of the stakeholders (i.e. intended). On the other hand, in relation to the requirements validation step (step 3.1.D), the proposed mapping index will facilitate the translation of user's expectations into a technical language. Besides, this index helps overcoming the problem of communicational filters (Section 3.1.C). The mapping index will be discussed in more details in Section 4.

(2) Usage context definition: an iterative process

The RDP is inherently iterative (Figure 1) and the RE tasks include a requirements management activity responsible of the evolution of requirements over time. Our iterative approach suggests that requirements evolution needs to be strengthened by the evolution of the contextual elements we already defined: this will help avoiding requirement misinterpretation and the resulting risks of ill-defined specifications.

Our approach is purpose-oriented. In fact, understanding the usage context (Section 3.1.B) concerns primarily the reasons certain choices and constraints for behavior, content and structure are made [30]. At that step (i.e. understanding the usage context), the focus is not yet on the technical aspects, features, operations, or components. Modeling with a purpose-oriented perspective leads to considering the opportunities stakeholders seek out to identify potential inappropriate usages to avoid.

Furthermore, iterating over domain concepts during the design process (Figure 2) may help both experts and designers to learn more about the possible usages of the GIS-to-be. Thus, it provides a deeper understanding of the usage context. It also helps detecting risks, new requirements and improving existent ones.

In the next Section, we will focus on associating the components of our approach to their respective elements of the RE and the RDP. We will then detail technologically some of our suggestions.

4 Recommendations for the Description of Intention of Use

With regard to the analysis performed in the previous sections, we propose hereafter a set of suggestions in order to help end-users describing their intentions of use. Referring to the steps of the RDP (see Figure 1), a list of these suggestions is presented in the last column of the Table 1 and detailed in the next paragraphs. A particular attention is paid to relate "user-contributed actions" to the RE guidelines and the corresponding RDP steps (see Table 1):

Table 1. User contribution in describing intentions of use

RDP Steps	RE guidelines	User contributed actions
1. Elicitation	A. capturing the purposes of users.	α
	B. understanding the geospatial data context	β
2. Analysis	C. modeling, analyzing & negotiating requirements	γ
3. Specification		
4. Validation	D. validation of requirements	

α During the requirement elicitation, end-users contribute collaboratively with their application domain design team to describe satisfactory or/and unsatisfactory conditions and criteria specific to the usage context. We implemented a Web 2.0 Collaborative Platform to illustrate our approach (see Figure 3). Requirements for satisfactory or/and unsatisfactory conditions may be defined directly by the end-users and their representatives. User-defined context of requirements extend the ranges of acceptable and unacceptable behaviors and help designers considering new alternatives. These alternatives may have been missed if the user would not have been involved. For example, an end-user may suggest that using the proposed road-network in the way it is semantically defined and topologically-structured will satisfy their needs for snow removal but will lead to faulty responses for route optimization in case of emergency (e.g. for fire trucks) because some alternative paths are not included in the GIS (e.g. service roads along highways and between highway lanes; large public parking lots that may help avoid traffic jams at street lights). Similarly, an end-user may notice that private roads and streets are not included in the proposed GIS application and influence the results of their planned traffic analysis; they may even suggest to add only the intersections with such roads and streets, without entering the complete roads and streets, as it would be enough for their needs but that it couldn't be satisfactory for emergency purposes. Without such information, experts may believe they had a robust GIS application design because they had a pretty complete road network based on existing standards and adapted from a GIS vendor's road network template. The design which is built without such a collaborative needs definition and risk analysis may lead to inappropriate uses as it is more likely to include a lower number of pertinent hints.

User involvement helps providing useful information and facilitates capturing new requirements. It also helps to include context-sensitive warnings in GIS applications (e.g. [21]) and to prepare user manuals with appropriate usage warnings [23]. Such details regarding users' requirements should be formalized and recorded in the usage context previously defined (Section 3.2.1).

ß Research in the field of RE made some advancement in terms of user-centered design tools (e.g. URN[2]). Such tools would help understanding the intended usage context especially because they support expressing "intentional elements" such as goals, actors, links, tasks and properties [6].

For instance, graphical symbols are understandable by end-users and facilitate the communication of their purposes [29]. User-oriented graphical modeling tools (e.g. Perceptory[3], GRL[4]) and languages (e.g. PictograF) may be leveraged to elicit the usage intentions. For example, presenting a building to a user using a simple pictogrammic polygon may not satisfy his expectations. He may be looking for displaying the building's detailed corners that would be better represented by complex pictogrammic polygons or by a combination of simple pictogram and textual/graphic definition of detailed requirements. We defined in Figure 3 a component which represents Perceptory, a simple geospatial visual modeling tool appropriate for non-expert users which has a rich dictionary to express the details when needed.

Research on action-driven ontologies addressed "intentionality" with a focus on the operations to be performed [36]. Nevertheless, intentionality in this paper is addressed with respect to RE guidelines and with a focus on a wider concept: the usage purpose.

[2] URN (User Requirements Notation) is standardized by the International Telecommunication Union in 2008 for the elicitation, analysis, specification, and validation of requirements: `http://jucmnav.softwareengineering.ca/ucm/bin/view/UCM/WebHome,2012`

[3] Perceptory: `http://sirs.scg.ulaval.ca/perceptory/english/enewindex2.asp`

[4] GRL: Goal-oriented Requirement Language is a language for supporting goal and agent-oriented modelling and reasoning about requirements

γ Recording information about the relationship between the intentions of use (provided by end-users and their representatives) and the quality attributes (understandable by experts in geomatics) has been outlined above (see Section 3.2.1). We propose here a possible modeling format of the proposed association index. Modeling format could be, for example, XML schema or ontology [15]. The index may be presented to end-users and stakeholders through an appropriate representation (i.e. graphical vs. formal) in respect to their expertise level [14]. This would facilitate requirements derivation and helps focusing on aligning needs and specifications. In Figure 3, the index is represented by the “Mapping Index” component.

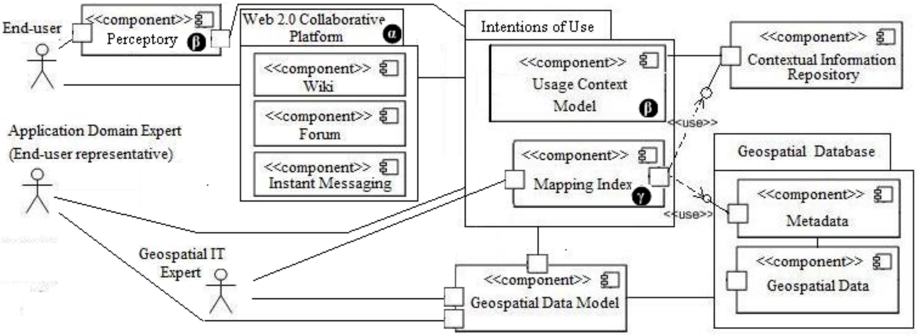


Fig. 3. Implementation example: a collaborative user-centered approach

Figure 3 illustrates an implementation of our Collaborative User-Centered Approach. The end-user has access to the Web 2.0 collaborative platform and defines his intentions of use through a Wiki, a Forum and an Instant Messaging system. He also, graphically, expresses his usage purposes using the graphical modeling tool (i.e. Perceptory). These elicitated requirements constitute the contextual information to be recorded with regards to the Usage Context Model. The recorded requirements are then mapped to their corresponding metadata forming what we call a Mapping Index.

Providing such facilities (i.e. mapping index, graphical modeling tools, and usage context model) facilitates early involvement of end-users in geospatial data design. In fact, these facilities allow a larger number of users to contribute to requirements definition (via the Web 2.0), provide them with an easy-to-use requirements representation tool to better define the scope of their needs (e.g. Usage Context Model) and help deriving specifications from these needs (e.g. Mapping Index) as well as context-sensitive warnings if desired and richer user manuals.

5 Conclusion

This paper presented an analysis of the geospatial data design process based on RE guidelines with the objective of outlining the steps where end-users involvement may be relevant in terms of detecting new requirements and improving existent ones. We presented, based on the RE sub-domain, an analysis of the role that end-users should play to help identifying and describing requirements. Referring to RE guidelines is important, since it establishes a set of principles to be respected in relation to user involvement in defining requirements. We then described how intentions of use and warnings about potential misuses can be captured and recorded, in terms of purposes, within a specific context model.

Our long-term research objective is to improve, through a participative process, the knowledge about risks related to inappropriate usage of geospatial data. The RE-based analysis performed in this paper shows that early involvement of end-users in geospatial data design by providing him with conceptual (e.g. usage context model) and technical (e.g. mapping index, user-centered and purpose-oriented graphical modeling tools) facilities is valuable for detecting new requirements and improving existing ones. It helped us to understand the impact of capturing and incorporating the intentions of use early in the design process. Future research would explore in detail the different aspects of the iterative process and its role in detecting risky issues.

Acknowledgments. The authors wish to acknowledge the contribution of Canada NCE-GEOIDE projects IV-23 “Public Protection and Ethical Dissemination of Geospatial Data” and the NSERC Discovery Grant of the second author.

References

1. Ko, J., Abraham, R., Beckwith, L., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M.B., Rothemel, G., Shaw, M., Wiedenbeck, S.: The state of the art in end user Software engineering. ACM Computing Surveys (2011)
2. Sutcliffe, A.: User-Centered Requirements Engineering. Springer (2002)
3. Nuseibeh, B.A., Easterbrook, S.M.: Requirements Engineering: A Roadmap. In: Finkelstein, A.C.W. (ed.) The Future of Software Engineering. Computer Society Press (2000)
4. Cheng, H.C., Atlee, J.M.: Research Directions in Requirements Engineering. In: Briand, L., Wolf, A.L. (eds.) Future of Software Engineering. IEEE-CS Press (2007)
5. Berry, D., Kamsties, E.: Ambiguity in Requirements Specification. In: Perspectives on Software Requirements, ch. 2. Kluwer Academic Publishers (2004)
6. Jackson, D.: Software Abstractions: Logic, Language, and Analysis. MIT Press (2006)
7. Hull, E., Jackson, K., Dick, J.: Requirements Engineering, 3rd edn. Springer (2011)
8. Sikora, E., Tenbergen, B., Pohl, K.: Industry needs and research directions in requirements engineering for embedded systems. Requirements Engineering 17(1), 57–78 (2012)
9. Van, T., Lamsweerde, V.A., Massonet, P., Ponsard, C.: Goal-oriented requirements animation. In: Proc. of the IEEE Int. Req. Eng. Conf (RE), pp. 218–228 (2004)
10. Sharp, H., Finkelstein, A., Galal, G.: Stakeholder identification in the requirements engineering process. In: Proc. of the 10th Int. Work. on Data. & Expert Syst. App. (1999)
11. Chandler, A., Levitt, K.: Spatial Data Quality: The Duty to Warn Users of Risks Associated with Using Spatial Data 49(1), 79–106 (2011), Alberta L. Review
12. Chandler, A.: Harmful Information: Negligence Liability for Incorrect Information. Short Note in R. Devillers and H. Goodchild, SDQ: From Process to Decisions. CRC Press (2010)

13. Brodeur, J., Bédard, Y., Edwards, G., Moulin, B.: Revisiting the Concept of Geospatial Data Interoperability within the Scope of Human Communication Processes. Transactions in GIS 7(2), 243–265 (2003)
14. Grira, J., Bédard, Y., Roche, S.: Spatial Data Uncertainty in the VGI World: going from Consumer to Producer. Geomatica, Jour. of the Can. Inst. of Geomatics 64(1), 61–71 (2009)
15. Omoronyia, I., Sindre, G., Stålhane, T., Biffl, S., Moser, T., Sunindyo, W.: A Domain Ontology Building Process for Guiding Requirements Elicitation. In: Wieringa, R., Persson, A. (eds.) REFSQ 2010. LNCS, vol. 6182, pp. 188–202. Springer, Heidelberg (2010)
16. ISO-TC/211. Geographic Information - Quality principles 19113 (2002)
17. Wiegers, K.E.: Software Requirements, 2nd edn. Microsoft Press (2003)
18. Pohl, K.: Requirements engineering - fundamentals, principles, techniques. Springer (2010)
19. Ryan, K.: The role of natural language in requirements engineering. In: Proceedings of the IEEE International Symposium on Requirements Engineering, San Diego, CA, pp. 240–242. IEEE Computer Society Press, Los Alamitos (1993)
20. Wasson, K.S.: A case study in systematic improvement of language for requirements. In: Proc. of the IEEE Int. Req. Eng. Conf. (RE), pp. 6–15 (2006)
21. Levesque, M.-A., Bédard, Y., Gervais, M., Devillers, R.: Towards managing the risks of data misuse for spatial datacubes. In: Proc. of the 5th ISSDQ, Enschede, Netherlands, June 13-15 (2007)
22. Aoyama, M.: Persona-and-scenario based requirements engineering for software embedded in digital consumer products. In: Proc. of the IEEE Int. Req. Eng. Conf., pp. 85–94 (2005)
23. Gervais, M.: Élaboration d'une stratégie de gestion du risque juridique découlant de la fourniture de données géographiques numériques. PhD thesis, Laval Univ (2003)
24. Kitamura, M., Hasegawa, R., Kaiya, H., Saeki, M.: A Supporting Tool for Requirements Elicitation Using a Domain Ontology. Comm. Computer & Info Sci. 22(pt. 2) (2009)
25. Sabetzadeh, M., Easterbrook, S.: Traceability in viewpoint merging: a model management perspective. In: Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering, pp. 44–49 (2005)
26. Feather, M.S.: Towards a unified approach to the representation of, and reasoning with, probabilistic risk information about software and its system interface. In: Int. Sym. on Soft. Reliab. Eng., pp. 391–402 (2004)
27. Shibaoka, M., Kaiya, H., Saeki, M.: GOORE: Goal-Oriented and Ontology Driven Requirements Elicitation Method. In: Hainaut, J.-L., Rundensteiner, E.A., Kirchberg, M., Bertolotto, M., Brochhausen, M., Chen, Y.-P.P., Cherfi, S.S.-S., Doerr, M., Han, H., Hartmann, S., Parsons, J., Poels, G., Rolland, C., Trujillo, J., Yu, E., Zimányie, E. (eds.) ER Workshops 2007. LNCS, vol. 4802, pp. 225–234. Springer, Heidelberg (2007)
28. Maiden, N., Robertson, S.: Integrating creativity into requirements processes: experiences with an air traffic management system. In: Proc. of the IEEE Int. Req. Eng. Conf. (2005)
29. Chen, S., Li, Y.: Visual Modeling and Representations of Spatiotemporal Transportation Data: An Object-Oriented Approach. In: ISCSS, pp. 218–222 (2011)
30. Faily, S.: Bridging User-Centered Design and Requirements Engineering with GRL and Persona Cases. In: Proceedings of the 5th International i* Workshop, pp. 114–119 (2011)
31. Konrad, S., Gall, M.: Requirements Engineering in the Development of Large-Scale Systems. In: Proc. of the 16th IEEE Int. Requirements Engineering Conf., September 8-12 (2008)

32. Asnar, Y., Giorgini, P., Mylopoulos, J.: Goal-driven Risk Assessment in Requirements Engineering. In: Requirements Engineering (2011)
33. Bédard, Y.: Data Quality + Risk Management + Legal Liability = Evolving Professional Practices. In: Proc. FIG Working Week, Marrakech, Morroco, May 16-22 (2011)
34. Y. Bédard, J. Chandler, R. Devillers, M. Gervais. System Design Methods and Geospatial Data Quality, Association of American Geographers. Professional Ethics-Session on Geographic Information Ethics and GIScience, March 22-27, Las Vegas, USA, 2009.
35. Robinson, W.N., Pawlowski, S.D., Volkov, V.: Requirements interaction management. ACM Computing Surveys 35(2), 132–190 (2003)
36. Câmara, G., Monteiro, A.M., Paiva, J., Souza, R.C.M., Miguel, A., Monteiro, V., Paiva, J.A., Cartaxo, R., Souza, M.D.: Action-Driven Ontologies of the Geographical Space: Beyond the FieldObject Debate. In: First International Conference on Geographic Information Science (2000)

Using the DBV Model to Maintain Versions of Multi-scale Geospatial Data

João Sávio C. Longo, Luís Theodoro O. Camargo, Claudia Bauzer Medeiros, and André Santanchè

Institute of Computing (IC) - University of Campinas (UNICAMP)
Campinas, SP – Brazil
{joaosavio,theodoro}@lis.ic.unicamp.br, {cmbm,santanche}@ic.unicamp.br

Abstract. Work on multi-scale issues concerning geospatial data presents countless challenges that have been long attacked by GIScience researchers. Indeed, a given real world problem must often be studied at distinct scales in order to be solved. Most implementation solutions go either towards generalization (and/or virtualization of distinct scales) or towards linking entities of interest across scales. In this context, the possibility of maintaining the history of changes at each scale is another factor to be considered. This paper presents our solution to these issues, which accommodates all previous research on handling multiple scales into a unifying framework. Our solution builds upon a specific database version model – the multiversion MVDB – which has already been successfully implemented in several geospatial scenarios, being extended here to support multi-scale research. The paper also presents our implementation of of a framework based on the model to handle and keep track of multi-scale data evolution.

Keywords: multi-scale, database versions, MVDB model.

1 Introduction

A major challenge when dealing with geospatial data are the many scales in which such data are represented. For instance, national mapping agencies produce multi-scale[1] geospatial data and one of the main difficulties is to guarantee consistency between the scales [15]. Most research efforts either concentrate on modeling or on data structures/database aspects.

Literature on the management of geospatial data at multiple scales concentrates on two directions: (a) generalization algorithms, which are mostly geared towards handling multiple spatial scales via algorithmic processes, that may, for instance, start from predefined scales, or use reactive behaviors (e.g., agents) to dinamically compute geometric properties; and (b) multi-representation databases (MRDBs), which store some predefined scales and link entities of interest across scales, or multiple representations within a scale. These two approaches roughly

[1] Unless specified, this paper uses the term “scale” to refer to cartographic scale.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 284–293, 2012.

correspond to Zhou and Jones' [16] multi-representation spatial databases and linked multiversion databases[2].

While generalization approaches compute multiple virtual scales, approaches based on data structures, in which we will concentrate, rely on managing stored data. From this point of view, options may vary from maintaining separate databases (one for each scale) to using MRDBs, or MRMS (Multiple Representation Management Systems) [5]. MRDBs and MRMS concern data structures to store and link different objects of several representations of the same entity or phenomenon [13]. They have been successfully reported in, for instance, urban planning, or in the aggregation of large amounts of geospatial data and in cases that applications require data in different levels of detail [8,7,10]. Oosterom et al. [9], in their multi-representation work, also comment on the possibility of storing the most detailed data and computing other scales via generalization. This presents the advantage of preserving consistency across scales (since all except for a basis are computed). Generalization solutions vary widely, but the emphasis is on real time computation, which becomes costly if there are continuous updates to the data – e.g., see the hierarchical agent approach of [12] or the multiple representations of [1].

This paper presents our approach to manage multiple scales of geospatial objects that is based on extending the DBV (*Database Version*) model [3,6] to provide support to flexible MRDB structures. As will be seen, our extension (and its implementation) provide the following advantages to other approaches: (a) it supports keeping track of evolution of objects at each scale, and across scales, simultaneously; (b) it provides management of multi-scale objects saving storage space [3], as opposed to approaches in which evolution requires replication; and (c) it supports evolution according to scale and to shape, where the latter can be treated as alternative versioning scenarios.

2 Basic Conceps and Related Work

2.1 MRDB and Multi-scale Data

Spaccapietra et al. [14] cite that in different scales the objects are usually represented in different ways, because each scale can have a convention of representation. Objects can appear/disappear or be aggregated/disaggregated, shapes can be simplified or objects could not appear in some scales.

Relying of this fact, MRDBs (Multiple Representation Database) have been proposed to solve this problem. These are data structures to store and link different objects of several representations of the same entity or phenomenon [13]. There are plenty of benefits to this approach, according to Sarjakoski [13]:

- Maintenance is flexible, since more specific level updates can be propagated to the lower resolution data;

[2] We point out that our definition of *version* is not the same as that of Zhou and Jones.

- The links between objects of different levels of representation can provide a basis for consistency and automatic error checking;
- MRDBs can be used for multi-scale analysis of spatial information, such as comparing data at different resolution levels.

According to Deng et al. [4], there are three main variants to link objects in an MRDB. The first one is called *attribute variant* and all data are stored in one dataset. The second variant, named *bottom-up variant*, considers the existence of two or more datasets, linked by an additional attribute that links the objects of the actual scale to those of the immediately smaller scale. The *top-down variant*, the third approach, is similar to the second, except for the fact that the link points to the immediately larger scale.

As an example of implementation, Parent et al. [11] present MurMur, an effort to develop a manipulation approach to geographic databases that have multiple representations. Additional research on MRDB structures includes Burghardt et al.'s work [2], which shows how to improve the creation of maps via automated generalization for topographic maps and multi-representation databases.

Although MRDB structures are used to treat multi-representation problems, this paper proposes to deal with multi-scale problems, a subset of those related to multi-representations. Our proposal allows keeping the history of changes within and across scales, which is not directly supported by MRDBs.

2.2 The DBV Model

The DBV (*Database Version*) model is an approach to "maintain consistency of object versions in multiversion database systems" [3]. A DBV represents a possible state or version of the database [3]. It can be seen as a virtual view of a database in which multiple versions of objects are stored. This view shows just one version of each object, so that users can work at each DBV as if they were handling a constant (monoversion) state of the database. Temporal versioning is just one type of version. Database researches and, more specifically, the DBV model, consider a version to be any stored modification of a (database) state. Thus, a given real world object mat be versioned in time, but also different simultaneous representations are versions of that object.

In this model, there are two levels: the logical and the physical. The first corresponds to the user view of each database state (DBV) and is represented by the *logical versions*. The second is represented by the *physical versions* of the stored objects.

A *multiversion object* represents one single entity in the real world – any attribute (geometry, color, etc) may change; as long as the experts consider it to be the same entity, it is not assigned a new *id*. Let us consider a *multiversion object o*, e.g. a car, with two different models, one painted blue and other red. Internally, the database will store the *physical versions* of *o* as *pv1* and *pv2*. Logically, *pv1* will appear in one DBV and *pv2* in another. The physical database will have cars of both colors, but from a logical (user's) point of view, only one color exists.

Versions are organized in a derivation tree as Figure 1 shows. Each version is associated with a *stamp* value (0, 0.1, etc). The derivation tree indicates how DBVs are derived from each other, thus supporting change traceability. Derivations always correspond to some kind of update. For instance, Figure 1 shows that DBV *d1* (*stamp* 0.1) is derived from *d0* (*stamp* 0) and that *d2* (*stamp* 0.1.1) and *d3* (*stamp* 0.1.2) are derived from *d1*. By definition, there is no data in *stamp* 0 (root).

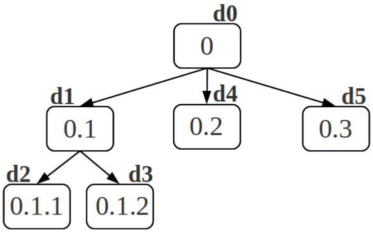


Fig. 1. Derivation tree of database versions

One of the main advantages of using the DBV approach is that only the changes must be stored. Data that are not modified are shared from previous DBVs through semantics of the version *stamps*. For instance, suppose we have to access all *logical versions* related to *d2*. It is also necessary to look up at all previous DBVs up to the root – *d1*, since each version stores only the data changes. More information about the DBV model can be seen in [3,6].

3 Our Approach

3.1 Overview

We have adopted the DBV model to support multiple scales. Each DBV represents a particular scale. The set of DBVs, which can be interlinked, correspond to a multi-scale/multi-representation world.

We extended the model so that, instead of one derivation tree, each scale has its own tree and all trees evolve together. Besides the version *stamp*, each DBV has an associated scale *s*. We use the following notation: the DBV *d0* of scale 1 as $d0_1$. Figure 2 shows four versions (0, 0.1, 0.1.1 and 0.1.2) and *n* scales.

Let a real world object *o1* be physically stored in a database in two scales, receiving physical identifiers *pv1* and *pv2*, where *pv1* is a polygon and *pv2* a point. Polygon and point are respectively represented in DBVs $d1_1$ and $d1_2$. Using this information and the DBV concepts, we have two *logical versions* (each in a DBV) represented in the following way: *logical version 1* = ((*o1*, $d1_1$), *pv1*) and *logical version 2* = ((*o1*, $d1_2$), *pv2*). In other words, DBV $d1_1$ contains the polygon version of *o1*, and $d1_2$ the point version of *o1*.

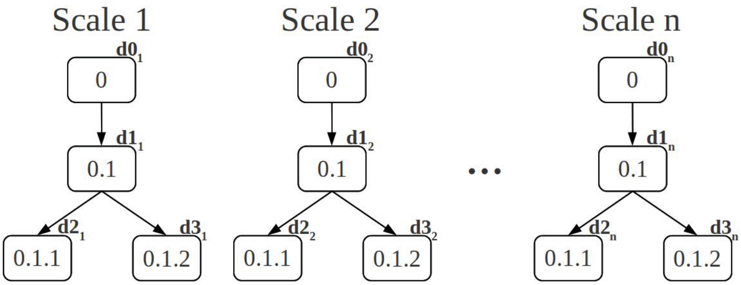


Fig. 2. Example of our approach to maintain versions of multi-scale geospatial data

Unlike several multi-representation approaches, we do not link explicitly objects of different scales (e.g., *pv1* and *pv2*). Instead, the link is achieved implicitly by combining *stamp* and derivation trees, using the concept of *logical versions*. This kind of link is similar to the *bottom-up variant* seen in section 2.1.

A change in a real world that requires creating a new version in scale s may require changes in other scales. Keeping one tree per scale, moreover, makes sense because, as remarked by [14], for large scale changes an object suffers radical changes when scale changes occur and thus there is seldom any intersection (if any) between DBVs in different scales. To simplify maintaining consistency across scales, we postulate that all derivation trees grow and shrink together and have the same topology. This leads to the notion of multi-scale *scenario* σ, for short, *scenario*. Each *scenario* is formed by all the DBVs with the same version *stamp*. For instance, in Figure 2, $d0_1$, $d0_2$, ..., $d0_n$ form a *scenario*, and so do $d1_1$, $d1_2$, ..., $d1_n$; etc. In fact, there may be many *scenarios*.

For managing the versions, we use the propagation algorithm adopted by the DBV model: only data changes must be stored and unchanged data are propagated across versions.

3.2 The Model

Figure 3 represents our model in UML. We introduce a new class called *Scale*, which has an identifier named *sid* (*scale id*). A DBV is identified by the couple (*stamp*, *sid*). The *Scale* class allows the association of a DBV with different types of scales, where spatial scale is one of them (another example is the temporal scale[3]). *LogicalVersion* class associates a *MultiversionObject* to a *DBV*. A *physical version* of an object underlies a *logical version* (i.e., it may appear in some DBV). This is expressed by the relationship between *LogicalVersion* and *PhysicalVersion* classes. The latter is the root of a hierarchy of classes of all kinds of objects that can be versioned (see Figure 5 later on) and allows the user to choose which data will be versioned though the *OBJTYPE* parameterized type. This approach forces the subclasses of *PhysicalVersion* to provide the data to

[3] This paper is restricted to spatial aspects.

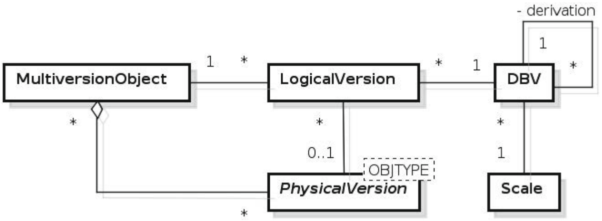


Fig. 3. Our basic model in UML

be versioned. If a *multiversion object o* does not appear in DBV *d*, we represent this situation setting the *PhysicalVersion* as *null*. A DBV has one parent and – by a derivation process – one or more children.

The model considers the following operations: (a) Create, modify and delete a *multiversion object* and its *physical versions*; (b) Create a new *Scale* (which will create a new tree); (c) Create or remove a DBV (affecting the trees); (d) Access a DBV (gathering the relevant objects of interest).

4 Implementation Details

4.1 Overview

We chose to implement our framework in an object-relational database due its widespread adoption and to its support of geospatial features. We developed an API[4] on top of the PostGIS[5] spatial database extension for PostgreSQL[6]. Our implementation uses the Java programming language, Java Persistence API (JPA)[7] and Hibernate Spatial[8] for geographic data object/relational mapping.

Figure 4 shows the architecture of the API, divided in three layers: *Domain Data Mapping*, *Handlers* and *Controller*. The *Domain Data Mapping* layer implements the database for the model of Figure 3, mapping Java objects into the underlying DBMS. The *Handlers* layer access the physical storage. This layer is inspired in the DAO (Data Access Object)[9] pattern, to which we added specific methods of our model. There are five handlers, each of which related to an entity of the *Domain Data Mapping* layer. The *Controller* layer is accessed by applications to select the DBV to use and to perform operations on.

[4] http://code.google.com/p/dbv-ms-api

[5] http://postgis.refractions.net

[6] http://www.postgresql.org

[7] http://jcp.org/aboutJava/communityprocess/final/jsr317/index.html

[8] http://www.hibernatespatial.org

[9] http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

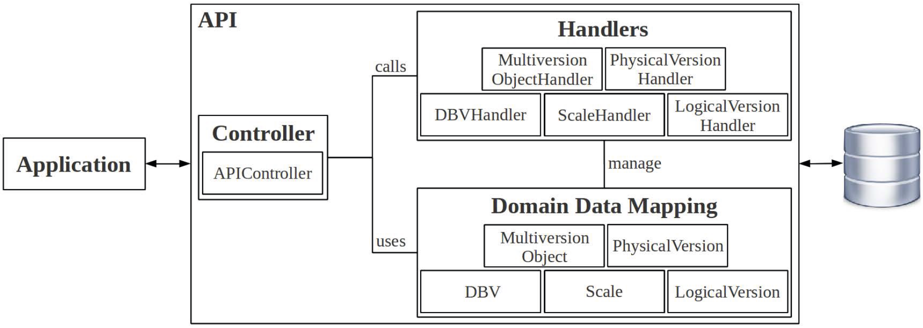


Fig. 4. Architecture of the API

4.2 Using the API

This section shows the steps for an application A to use the API to create multi-scale databases.

Step 1 - Create Subclasses of *PhysicalVersion*. First of all, it is necessary to create the schema, i.e., subclasses of *PhysicalVersion*, binding the appropriate parametrized type, which will indicate the *OBJTYPE* to be versioned. Figure 5 shows two examples of subclasses of *PhysicalVersion*. Both have the same attributes, but different versioned data (because of the different binding parametrized type). The *GeometryPV* subclass is versioning only the *spatialObject* attribute while *SpatialPV* subclass is entirely versioned. Each subclass created by the user will represent a different table in the multiversion database.

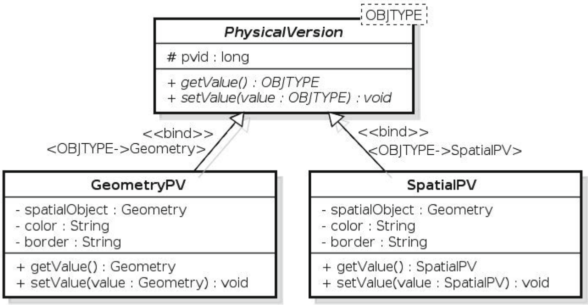


Fig. 5. Examples of subclasses of *PhysicalVersion*

Step 2 - Add Data. In order to add new data, the first step will be to select a specific version *stamp* of a DBV. Here, A inserts *multiversion objects* and their *physical versions*. A *MultiversionObject* class has three attributes: *oid*, *title* and a list of *PhysicalVersions*. The first is the identifier, the second is some title which identifies the object in the real world, and the third represents the associated *physical versions* plus their scales. Also, it is necessary to define the spatial scales to be available. Every time a *Scale* is added, a new derivation tree is created (by creating a root DBV). A *Scale* class has three attributes: *sid*, *type* and *value*. The first is the identifier, the second represents the type of the scale: spatial, temporal, etc, and the third attribute is the value associated to the *type*. For instance, for spatial scales, the *value* contains their size (e.g. 1:10000).

Step 3 - Perform Operations. Once steps 1 and 2 are performed, applications can invoke operations on objects and DBVs, via invocation of methods of the *Controller* layer, e.g., adding, removing and updating the *logical versions*, by working in a scale at a time. Now, suppose we have already done the changes and we want to make a new version of them. When we create a new DBV from the current, the changes are saved. Subsequent versions can be built by changing the *logical versions* and creating new DBVs.

Consider Figure 6, where the roots (*stamp* 0) appear for scales 1:10000, 1:20000 and 1:50000. This example concerns urban vectorial data, and the Figure illustrates a given city section. The first version from root (*stamp* 0.1) shows the initial state of the section represented in the three scales. Version 0.1.1 and 0.1.2 show evolution alternatives in that section (either prioritizing the horizontal road, or the vertical road). The geometries in dotted lines represent the propagated data.

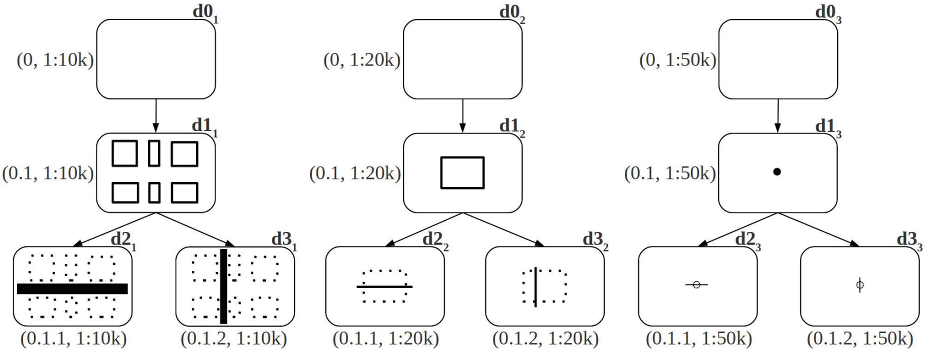


Fig. 6. Multi-scale versioning problem example

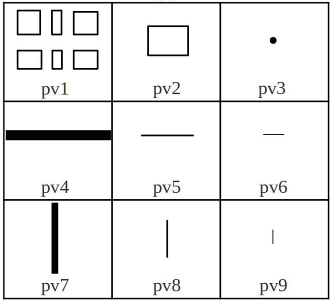
Internal details appear in Figure 7. Part (a) shows the *multiversion objects* and their *physical versions*. Part (b) shows the *physical versions* and their geometry. Finally, the *logical versions* and their relationship with *physical versions* are shown in part (c). For instance, in scale 1:10000, the city section is stored as a complex geometry (a polygon with six sub-polygons), with *oid o1* and with three physical representations – one per scale – *pv1*, *pv2* and *pv3*. Each of these geometries will be accessible via a different DBV, respectively $d1_1$, $d1_2$ and $d1_3$.

Suppose the user wants to work at scale 1:10000, in the horizontal road situation, i.e., DBV $d2_1$. The DBV view is constructed from all objects explicitly assigned to it (*pv4* of *o2*), and all objects in previous DBVs of that scale, up

(a)

MultiversionObject		
Title	**Oid**	**Physical Versions**
city section	o1	pv1 pv2 pv3
horizontal road	o2	pv4 pv5 pv6
vertical road	o3	pv7 pv8 pv9

(b)



(c)

LogicalVersion		
Multiversion Object	**DBV**	**Physical Version**
o1	$d1_1$	pv1
o1	$d1_2$	pv2
o1	$d1_3$	pv3
o2	$d2_1$	pv4
o2	$d2_2$	pv5
o2	$d2_3$	pv6
o3	$d3_1$	pv7
o3	$d3_2$	pv8
o3	$d3_3$	pv9

Fig. 7. (a) *Multiversion objects* and their *physical versions*. (b) *Physical versions* and their geometry. (c) *Logical versions* from the example.

to the root, i.e., $d1_1$ – *pv1* of *o1*. This construction of consistent scenarios for a given scale in time is achieved via the *stamps*, by the DBV mechanism. Notice that each version is stored only once. Unless objects change, their state is propagated through DBVs, saving space. Also, users can navigate across a path in the derivation tree, following the evolution of objects in time. For more details on space savings, see [3].

5 Conclusions and Future Work

We have presented an approach to manage multi-scale geospatial data, and keep track of their evolution, through the DBV model. This proposal was implemented in a prototype, developed in order to validate our solution. We have already implemented some toy examples, which show the advantages of this proposal, and are constructing a test suite with real data.

Our framework supports the traceability of the evolution of spatial objects, while at the same time handling multi-scale management. Thanks to the adoption of the DBV model, storage space is saved [3], and the separation between *physical* and *logical versions* facilitates the creation of consistent, single scale views over multi-scale data.

We point out that our approach is centered on data structures to store and manage multi-scale data. This allows controlling updates, keeping history of evolution in the real world and other issues that can be efficiently handled only in a storage based policy. Nevertheless, the DBV infrastructure can be used as a basis for any kind of generalization approach – e.g., construction of intermediate scales, generalization of alternative virtual scenarios, and so on, to work, for instance in digital cartography.

Future work includes versioning along the temporal scale and specification of integrity constraints across scales, to determine rules for update propagation.

References

1. Bédard, Y., Bernier, E., Badard, T.: Multiple representation spatial databases and the concept of vuel. In: Encyclopaedia in Geoinformatics. Idea Group Publishing, Hershey (2007)
2. Burghardt, D., Petzold, I., Bobzien, M.: Relation modelling within multiple representation databases and generalisation services. The Cartographic Journal 47(3), 238–249 (2010)
3. Cellary, W., Jomier, G.: Consistency of versions in object-oriented databases. In: Proc. of the 16th Int. Conference on Very Large Databases, pp. 432–441. Morgan Kaufmann (1990)
4. Deng, X., Wu, H., Li, D.: Mrdb approach for geospatial data revision. In: Proc. of SPIE, the Int. Society for Optical Engineering (2008)
5. Friis-Christensen, A., Jensen, C.: Object-relational management of multiply represented geographic entities. In: Proc. 15th Int. Conference on Scientific and Statistical Database Management SSDBM (2003)
6. Gançarski, S., Jomier, G.: A framework for programming multiversion databases. Data Knowl. Eng. 36, 29–53 (2001)
7. Gao, H., Zhang, H., Hu, D., Tian, R., Guo, D.: Multi-scale features of urban planning spatial data. In: 18th Int. Conference on Geoinformatics, pp. 1–7 (2010)
8. van Oosterom, P.: Research and development in geo-information generalisation and multiple representation. Computers, Environment and Urban Systems 33(5), 303–310 (2009)
9. van Oosterom, P., Stoter, J.: 5D Data Modelling: Full Integration of 2D/3D Space, Time and Scale Dimensions. In: Fabrikant, S.I., Reichenbacher, T., van Kreveld, M., Schlieder, C. (eds.) GIScience 2010. LNCS, vol. 6292, pp. 310–324. Springer, Heidelberg (2010)
10. Parent, C., Spaccapietra, S., Vangenot, C., Zimányi, E.: Multiple representation modeling. In: Encyclopedia of Database Systems, pp. 1844–1849. Springer US (2009)
11. Parent, C., Spaccapietra, S., Zimányi, E.: The murmur project: Modeling and querying multi-representation spatio-temporal databases. Information Systems 31(8), 733–769 (2006)
12. Ruas, A., Duchêne, C.: Chapter 14 - a prototype generalisation system based on the multi-agent system paradigm. In: Generalisation of Geographic Information, pp. 269–284. Elsevier Science B.V. (2007)
13. Sarjakoski, L.T.: Chapter 2 - conceptual models of generalisation and multiple representation. In: Generalisation of Geographic Information, pp. 11–35. Elsevier Science B.V. (2007)
14. Spaccapietra, S., Parent, C., Vangenot, C.: GIS Databases: From Multiscale to MultiRepresentation. In: Choueiry, B.Y., Walsh, T. (eds.) SARA 2000. LNCS (LNAI), vol. 1864, pp. 57–70. Springer, Heidelberg (2000)
15. Stoter, J., Visser, T., van Oosterom, P., Quak, W., Bakker, N.: A semantic-rich multi-scale information model for topography. Int. Journal of Geographical Information Science 25(5), 739–763 (2011)
16. Zhou, S., Jones, C.B.: A Multirepresentation Spatial Data Model. In: Hadzilacos, T., Manolopoulos, Y., Roddick, J., Theodoridis, Y. (eds.) SSTD 2003. LNCS, vol. 2750, pp. 394–411. Springer, Heidelberg (2003)

Multi-scale Windowing over Trajectory Streams

Kostas Patroumpas

School of Electrical and Computer Engineering
National Technical University of Athens, Hellas
kpatro@dbnet.ece.ntua.gr

Abstract. Many modern monitoring applications collect massive volumes of positional information and must readily respond to a variety of continuous queries in real-time. An important class of such requests concerns evolving trajectories generated by the streaming locations of moving point objects, like GPS-equipped vehicles, commodities with RFID's etc. In this paper, we suggest an advanced windowing construct that enables online, incremental examination of recent motion paths at multiple levels of detail. This spatiotemporal operator can actually abstract trajectories at progressively coarser resolutions towards the past, while retaining finer features closer to the present. We explain the semantics of such multi-scale sliding windows through parametrized functions that can effectively capture their spatiotemporal properties. We point out that window specification is much more than a powerful means for efficient processing of multiple concurrent queries; it can be also used to obtain concrete subsequences from each trajectory, thus preserving continuity in time and contiguity in space for the respective segments. Finally, we exemplify window utilization for characteristic queries and we also discuss algorithmic challenges in their ongoing implementation.

Keywords: Geostreaming, Moving Objects, Trajectories, Windows.

1 Introduction

As location-based services gain popularity with the proliferation of smartphones and positioning devices (GPS, RFID, GSM), it becomes harder to sustain the bulk of rapidly accumulating traces from a multitude of vehicles, ships, containers etc. Monitoring applications usually focus on current positions and spatial relationships amongst such moving objects, but the significance of their trajectories is rather overlooked. Still, continuous tracking of mobile devices offers an evolving trace of their motion across time. As numerous objects may relay their locations frequently, a vast amount of positional information is being steadily accumulated in a streaming fashion [13], rendering all the more difficult its real-time processing in main memory. So, user requests naturally focus on the latest portion of data through repeatedly refreshed *sliding windows* [12,14] that span a recent time interval (e.g., only inspect data received over the past hour).

In such a *geostreaming* context, we argue that the significance of each isolated position in a trajectory is time-decaying, until it eventually becomes obsolete

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 294–303, 2012.

and practically negligible. Taking inspiration from such an "amnesic" behavior [13], in this paper we introduce the notion of *multi-scale sliding windows* against trajectory streams. Instead of just restricting the focus on recent past, we extend our previous work on multi-granular windows at varying levels of detail [11] to exploit spatiotemporal properties inherent in evolving trajectories. We deem that windowing can effectively retain several, gradually coarser representations of each object's movement over greater time horizons towards the past; in return, higher precision should be reserved for the most recent segments. This can be achieved through diverse *scaled* representations per time horizon, in order to obtain increasingly generalized, yet comparable traces for all objects, no matter their actual reporting frequency. For example, consider an application that monitors delivery trucks for a logistics firm. Apparently, finest "zigzag" details of each itinerary mostly matter for the latest 15 minutes. Suppressing most of them could still reliably convey motion characteristics over the past hour, whereas just a few waypoints suffice to give the big picture throughout the day. Hence, this operator acts as an online simplifier per trajectory and incrementally maintains multiple representations at prescribed resolutions and time periods.

In practice, this idea may be proven advantageous for applications like fleet management, traffic surveillance, wildlife observation, merchandise monitoring, maritime control, soldier tracking in battlefields etc. Typical operations include:

- *Trajectory filtering* to facilitate range or k-NN search, by examining contemporaneous, lightweight portions of trajectories at comparable scales.
- *Ageing trajectory synopses*, smoothly updated with time and gracefully compressed with age, retaining only the most salient spatiotemporal features.
- *Efficient motion mining* to identify recent trends at varying resolutions.
- *Online multi-grained aggregates* per trajectory, e.g. heading, speed, etc.
- *Advanced visualization* of trajectory features on maps at diverse zoom levels.

To the best of our knowledge, this is the first attempt to introduce composite windows over streaming trajectories. Here is a summary of our contributions:

(i) We advocate for multi-scale sliding windows as a means of capturing essential motion features from the evolving geostream, and we discuss their parametrized semantics in space and time (Section 3).

(ii) We designate maintenance methods to ensure cohesion of trajectory segments using a series of common articulation points that leave no gaps between trajectory features at successive window levels (Section 4).

(iii) We indicate that typical spatiotemporal predicates and functions are directly applicable to these alternate, compressed representations. In addition, we demonstrate the efficacy of multi-scale windowing through SQL-like expressions for typical continuous queries involving trajectories (Section 5).

2 Background

As our work attempts to fuse ideas from window-based stream processing and multi-granular semantics into trajectory management, we next discuss fundamental concepts and related work in these domains.

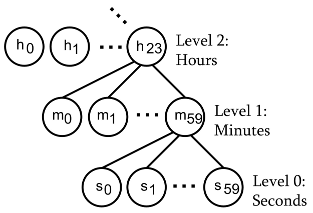
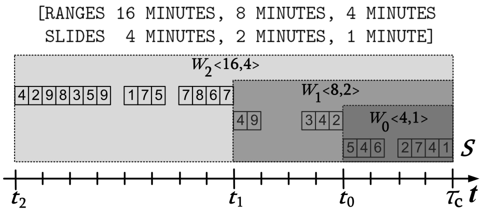


Fig. 1. Hierarchical time granules



```
[RANGES 16 MINUTES, 8 MINUTES, 4 MINUTES
 SLIDES  4 MINUTES, 2 MINUTES, 1 MINUTE]
```

Fig. 2. State of a 3-level sliding window

Windows over Data Streams. Continuous query execution has been established as the most renowned paradigm for processing transient, fluctuating and possibly unbounded *data streams* [14] in many modern applications, like telecom fraud detection, financial tickers or network monitoring. In order to provide real-time response to numerous *continuous queries* that remain active for long, most processing engines actually restrict the amount of inspected data into temporary, yet finite chunks. Such windows [12] are declared in user requests against the stream through properties inherent in the data, mostly using timestamping on incoming items. Typically, users specify *sliding windows*, expressing interest in a recent time period ω (e.g., items received during last 10 minutes), which gets frequently refreshed every β units (e.g., each minute), so that the window slides forward to keep in pace with newly arrived tuples. At each iteration, the temporary *window state* consists of stream tuples within its current bounds; usually $\beta < \omega$, so successive window instantiations may share tuples (state overlaps).

Multi-granular Semantics. Apparently, the sliding window paradigm dictates a single timeline of instants at similar detail. Yet, time dimension naturally adheres to a hierarchical composition of *time granules*, i.e., multiple levels of resolution with respect to Time Domain $\mathbb{T}$. Each granule γ_k at level k consists of a fixed number of discrete instants $\tau \in \mathbb{T}$. Merging a set of consecutive granules at level k leads into a greater granule at $k + 1$, thus iteratively defining several levels of *granularity* [2], like seconds, minutes, hours etc. as depicted in Fig. 1.

In spatiotemporal databases, functionality to support semantic flexibility of multiple representations and cartographic flexibility at multiple map scales was proposed in [10]. Representations of a real-world phenomenon may vary according to the chosen perception, i.e., time, scale, user profile, point of view etc. A formal model for multi-granular types, values, conversions and queries has been developed in [1], also handling evolutions due to dynamic changes and events.

Trajectory Management. Several models and algorithms have been proposed for managing continuously moving objects in spatiotemporal databases. In [7], an abstract data model and query language are developed towards a foundation for implementing a spatiotemporal DBMS extension where trajectories are considered as moving points. Based on that infrastructure, the SECONDO prototype [8] offers several built-in and extensible operations. Besides, a discrete model proposed in [5] decomposes temporal development into fragments, and then uses a simple function to represent movement along every "slice". For trajectories, each time-slice is a line segment connecting two consecutively recorded locations of a given object, as a trade-off between performance and positional accuracy. Interpolation can be used to estimate intermediate positions and thus approximately reconstruct an acceptable trace of the entire movement.

Multi-granular Window Processing. In [11] we introduced the notion of a multi-level sliding window W as a set of n time frames at diverse user-defined granularities. The goal was to concurrently evaluate a single continuous query over stream chunks of varying size. Subwindow W_k at level k has its own time *range* ω_k and *slide step* β_k. But each substate contains its subordinate ones, and all get nested under the widest W_{n-1} and keep up with current time τ_c, as exemplified in Fig. 2 for a 3-level window over stream S of integer values. A hierarchy of n subsumed frames is created when $\beta_{k-1} \leq \beta_k$ and $\omega_{k-1} < \omega_k$ for level $k = 1, \ldots, n-1$. For smooth transition between successive substates at any level k, we prescribed that $\omega_k = \mu_k \cdot \beta_k$ for $\mu_k \in \mathbb{N}^*$, so a given frame W_k consists of a fixed number of primary granules of size β_k units each. Emanating from this concept, we next propose a framework for online processing of streaming trajectories using windows at multiple temporal extents and spatial resolutions.

3 Multi-scale Sliding Windows over Evolving Trajectories

3.1 Rationale

In the sequel, we assume a discrete model with 3-d entities of known identities moving in two spatial and one temporal dimensions, i.e. point objects (not regions or lines) moving in Euclidean plane across time as illustrated in Fig. 3a. For a given object id, its successive samples $p \in \mathbb{R}^2$ are pairs of geographic coordinates (x, y), measured at discrete, totally ordered timestamps t from a given Time Domain $\mathbb{T}$ of primitive time instants (e.g., seconds). When a large number M of objects are being monitored, their relayed timestamped locations effectively constitute a *positional stream* of tuples $\langle id, p, t\rangle$ that keep arriving into a central processor from all moving sources. Thus, each trajectory is approximated as an evolving sequence of point samples collected from the respective data source at distinct time instants (e.g., a GPS reading taken every few seconds). Of course, sampling rates may not be identical for each object and may be varying even for a single object. However, no updates are allowed to already registered locations so that coherence is preserved among *append-only* trajectory segments.

Due to the sheer volume of positional updates and the necessity to answer continuous queries in real-time, it is hardly feasible to deal with lengthy, ever growing trajectories that represent every detail of the entire history of movement. Instead, it becomes imperative to examine lightweight, yet connected motion paths for a limited time period close to the present. Thanks to monotonicity of time, the semantics of sliding windows [12] against such positional streams are an ideal choice, as trajectories always evolve steadily along the temporal

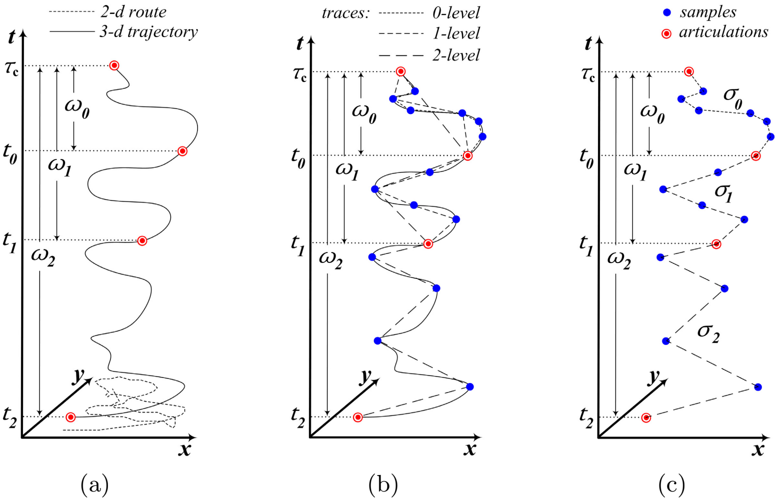


Fig. 3. Multi-scale sliding window over trajectory: (a) Original trajectory. (b) Traces over diverse time horizons. (c) Merging non-overlapping traces into a unified synopsis.

dimension. In effect, windows can abstract the recent portion of trajectories and thus provide dense subsequences without gaps.

For efficiency, we suggest that continuous queries could be evaluated against less detailed representations of objects' movement, purposely compressed on-the-fly. In a nutshell, the underlying semantics of the proposed windowing operator is *"drop detail with age"*. This data reduction refers to both spatial and temporal properties of trajectory features and is applied in an "amnesic" fashion per trajectory, after grouping positional updates by their respective object identifiers.

3.2 Semantics

The proposed window operator should act as a filter in two successive stages:

(i) it first narrows the examined streaming data down to finite chunks of reported locations at progressively smaller time intervals (*time-based filtering*);
(ii) then, it performs regulated generalizations against each retained subsequence of locations pertaining to the same object (*trajectory-based filtering*).

Time-Based Filtering. Suppose that each frame W_k has a fixed-size temporal extent (*range*) ω_k always greater than its subordinate ones, whereas it moves forward every β_k time units (its *slide* parameter). Assuming that all frames are firstly applied at time τ_0, the actual bounds of subwindow W_k at any time instant $\tau_c \geq \tau_0$ can be determined through its *scope*, i.e., a time interval

$$scope_k(\tau_c) = [\ \mathtt{max}\{\tau_0,\ \tau_c - \lambda_k - \omega_k + 1\},\ \tau_c - \lambda_k\ ]$$

where $\lambda_k = \texttt{mod}(\tau_c - \tau_0, \beta_k)$ is a time-varying lag behind current timestamp τ_c. Note that the rear bound of the scope for each W_k is initially τ_0, meaning that the subwindow is not yet filled as long as its range is less than ω_k. Afterwards, the rear bound becomes $t_k = \tau_c - \lambda_k - \omega_k + 1 > \tau_0$, while the front one is at $\tau_c - \lambda_k$. Both bounds slide by β_k units in tandem, once it holds that $\lambda_k = 0$. Hence, each frame neither slides forward at each timestamp nor upon arrival of every positional item, but discontinuously at distinct time instants in a deterministic, predictable pattern [12]. At every iteration of this stage after a slide, the set of qualifying items $\mathcal{C}_k(\tau_c) = \{s \in S : s.t \in scope_k(\tau_c)\}$ materializes the *time-filtered state* of respective subwindow W_k, e.g., positions taken over the past hour.

Trajectory-Based Filtering. Next, a *demultiplexing* step is employed for every frame W_k, so as to partition all positional tuples from $\mathcal{C}_k(\tau_c)$ into distinct paths of time-ordered positions according to their associated object identifiers. Hence, for any given object i, a sequence is obtained at each level k:

$$path_k(i) = \{s \in \mathcal{C}_k(\tau_c) : s.id = i \wedge (\forall s' \in \mathcal{C}_k(\tau_c), s'.id = i : s.t < s'.t \vee s'.t < s.t)\}.$$

In essence, a set $\mathcal{P}_k(\tau_c) = \{path_k(i), \forall i, 1 < i < M\}$ of truncated trajectories is derived, each spanning over the recent interval ω_k prescribed for the k^{th} window level. But the bulk of positional data destined for evaluation may still be considerable, especially if windows have large scopes and many levels. Therefore, a data reduction process is being applied against those truncated paths, also taking into account the actual detail of every original trajectory, as objects may not necessarily have a standard reporting frequency.

More specifically, let $|path_k(i)|$ signify the number of points reported from object i during past interval ω_k. As it may occur $|path_k(i)| \gg |path_k(j)|$ for two distinct objects i, j during the same interval ω_k, this filtering step should attempt to create comparable traces amongst trajectories with possibly diverse number of point samples. Given a maximum stream arrival rate ρ_{max} locations/sec, suppose that object i has an average rate $\rho_i \leq \rho_{max}$ of locations reported during ω_k. Therefore, a total of $\rho_i \cdot \omega_k$ samples have been recently relayed. Of them, at most $\delta_k^i \cdot \omega_k$ remain after randomly discarding or judiciously selecting point samples, where δ_k^i is an intended smoothing factor to be applied against that trajectory. This implies a reduction ratio of $\sigma_k = \frac{\delta_k^i}{\rho_i}$, in case that $\rho_i > \delta_k^i$. Otherwise, point elimination is not necessary, and all collected samples are retained intact.

By specifying a fixed ratio $\sigma_k < 1$ for all trajectories at level k, we can restrict the accuracy of their spatial representation up to a given detail. Effectively, σ_k acts as a *scale* parameter for frame W_k of the composite window, and prescribes the maximum degree of detail tolerated amongst its accumulated point positions per path. So, all trajectories should be separately smoothed at equivalent approximations, with several samples getting intentionally discarded (we discuss selection options later on). This does not necessarily mean that a similar number of point locations is finally retained per trajectory; depending on the actual course, much less than $\delta_k^i \cdot \omega_k$ points may constitute a path, e.g., when an object moves along a straight path with almost constant speed. But in general, the less the scaling factor at a given level k, the sparser the locations preserved per trajectory, so the various σ_k values per frame actually control the intensity of the applied approximation. Note that each such compressed representation still represents a connected $path'_k(i)$ consisting of those locations retained in the sequence (Fig. 3b), thanks to the inherent ordering of their timestamps.

Thus, the overall result is $\mathcal{W}_k(\tau_c) = \{path'_k(i), \forall i, 1 < i < M\}$. We emphasize that this *trajectory-filtered state* of window at level k includes a single compressed sequence per monitored object i, as opposed to dispersed timestamped positions that would have been returned by a common sliding window.

3.3 Discussion

Original trajectory data itself does not become multi-granular (e.g., alternate sequences in hours or days), and the underlying spatiotemporal model for their representation is kept as simple as possible. Instead, it is the proposed windowing operator that produces a series of n temporary datasets of increasing temporal extents and sparser positional samples. Those repetitively refreshed traces are meant to be utilized primarily in query evaluation, and not necessarily for permanent storage. Time range and scale parameters are defined by the users in their requests, thereby controlling the desired accuracy of query answers.

Granularities refer to levels of detail strictly for window specification and do not concern actual data representation. All data is modeled with a uniform schema including their spatial (position p) and temporal properties (timestamp t). Of course, relationships may exist among granularities, i.e., *'finer-than'* and *'coarser-than'* operators defined through inclusion and overlapping [1,2]. Thanks to such inherent relationships, most granules can be mapped onto the finest one supported by the model (e.g., seconds), and thus simplify calculations.

Multi-scale windowing should be clearly distinguished from *partitioned windows* [12], where attributes are also used to demultiplex incoming items into disjoint partitions. In our case, an additional "path creation" step is involved, which yields sequential data per object. Our policy also differs substantially from *load shedding* techniques applied against streaming locations [6]. Indeed, data points are not judiciously dropped upon arrival, but after being processed and digested into coarser traces. All relayed positions are admitted into the underlying stream database and become readily available for precessing. Then, each query may stipulate diverse time horizons and scales through windows, eventually discarding superfluous points and producing multi-resolution paths.

4 Towards Efficient Maintenance of Window States

It should be stressed that the proposed window operator does not solely extract simplified paths from trajectories, but should perform this task repetitively as trajectories evolve, while also offering multi-scale representations for querying. So, processing must be *incremental* as fresh object locations continuously arrive, and also *shared*, by exploiting already computed paths as much as possible.

Towards those intertwined goals, we opt for a strategy that can exploit point locations across many window frames. Since more detailed representations are prescribed for the narrowest frame (i.e., closest to present), selected point samples per trajectory may be progressively discarded when adjusting the compressed segments upwards in the window hierarchy. In effect, fewer and fewer points remain for the coarser frames by eliminating certain motion details, in accordance with their scale ratio σ_k. Preservation of certain *articulation points* per trajectory (Fig. 3b) is our seminal idea for a coordinated maintenance of multiple paths. Those points signify object locations at time instants that mark frame boundaries (or samples available closest to that time, depending on reporting frequency). Our intended method could promote a fair share of indicative locations to wider frames by keeping account of such points persistently; it may also yield a cohesive representation with non-overlapping point sets per level, each spanning consecutive intervals joined at those 'articulations' (Fig. 3c). Apart from expectations for optimized state maintenance, this scheme might also prove advantageous for a versatile portrayal of trajectories across multiple scales.

Retained samples from diverse trajectories may not be necessarily synchronized, i.e., measured at identical time instants. Although synchronization actually facilitates comparisons among trajectories, it might lead to oversimplified paths that could occasionally miss particular motion details when samples are chosen at a fixed frequency. Instead, samples should keep each compressed trace as much closer to its original course, chiefly by minimizing approximation error as in trajectory fitting [3,9]; we search for methods to achieve this in real-time.

Another crucial issue for state maintenance is how to attain a specific smooth factor σ_k per subwindow W_k. We have begun examining several alternatives:

Sampling. Typically, systematic sampling involves a random start and then proceeds with the selection of every m^{th} element from then onwards. In our case, the starting location may be the latest positional tuple arrived; then, randomly retain one sample from every successive batch of $\lceil \frac{\rho_i}{\delta_k^i} \rceil$ locations available per trajectory i at level k. Since this is a single-pass process (i.e., rewinding the trace backwards), the cost is $O(1)$ per location. On the downside, such a policy might occasionally lead to distorted or largely deviating traces for several objects.

Minimal Distance Errors. This strategy eliminates points that would incur the smallest change in the shape of each trajectory by minimizing synchronized distances, while preserving up to $\sigma_k \cdot \omega_k$ locations (bounded memory space per level). Yet, this spatiotemporal variant of Douglas-Peucker algorithm [9] could incur significant cost, as multiple passes may be needed per trajectory.

Online Filtering at Frame Transitions. By employing the maintenance scheme from [11] with a chain of core and buffer nodes, we could handle point selection only at transitions between frames (i.e., in buffer nodes). Involving a small fraction of the accumulated samples, such a process might gracefully drop less important samples in an incremental fashion when aging locations ascend through the stairwise organized frames. Thanks to inherent nesting, each frame needs to handle trajectory segments that are not already covered by its subordinate ones.

5 Perspectives

In this section, we briefly discuss the potential usage of multi-scale windowing in specifying and evaluating continuous queries involving trajectory streams.

By now, it should be clear enough that every window instantiation provides an updated set of recent paths per monitored object. Although trajectories span increasingly wider intervals in the past and may be compressed at diverse scales per level, the window state always offers contiguous, yet lightweight traces per object. This comes to the great benefit of an unobstructed evaluation of topological and spatiotemporal predicates, similarly to those established in [4,7].

Further, we can define auxiliary functions and predicates against such sequences in order to abstract particular aspects of the data. We particularly advocate for two functions, namely `trace()` and `trajectory()`, intended to return a linear series of point locations per moving object. The former reconstructs particular traces against each subwindow (Fig. 3b), whereas the latter yields a "merged" synopsis composed from successive multi-scale segments (Fig. 3c). Against the resulting timestamped polylines, we can then apply typical spatiotemporal functions (e.g., `speed`, `duration`) or predicates (like `INTERSECTS`, `CROSSES`, `WITHIN` etc.), always receiving meaningful, yet approximate answers.

Such language constructs may prove valuable for expressing several types of continuous queries over trajectories. Due to lack of space, we provide only two characteristic examples, assuming a positional stream `S <id, pos, ts>` of vehicle locations relayed into a traffic control center. Note that we expand the typical SQL clause for sliding windows [11] with additional terms concerning the prescribed scale factor per level and the distinguishing attribute of trajectories. Compared to multiple local views (one per subwindow) eventually combined into a `SELECT` statement, this concise rendition excels in expressiveness by far.

Example 1. "Estimate average speed of trajectories against varying time periods and scales". Function `WSCOPE(*)` actually indicates the respective scope for each computed subaggregate to annotate results properly:

```
SELECT AVG(speed(trace(S.pos))), WSCOPE(*)
FROM S [RANGES 1 HOUR, 15 MINUTES, 1 MINUTE
        SLIDES 5 MINUTES, 1 MINUTE, 10 SECONDS
        SCALES 0.1, 0.2, 0.5 BY S.id];
```

Example 2. "Approximately indicate vehicles circulating in Athens area recently."

```
SELECT S.id, duration(trajectory(S.pos))
FROM S [RANGES 30 MINUTES, 10 MINUTES, 1 MINUTE
        SLIDES 10 MINUTES, 1 MINUTE, 15 SECONDS
        SCALES 0.1, 0.2, 0.4 BY S.id],
     Cities C
WHERE trajectory(S.pos) WITHIN C.polygon
  AND C.name='Athens';
```

6 Concluding Remarks

In this paper, we set out the foundation for a novel windowing construct at multiple levels of detail against streaming trajectory data. We explained the semantics of such multi-scale sliding windows and presented certain interesting properties, which may enable their efficient shared evaluation. We also introduced language constructs and exemplified their usage in spatiotemporal continuous queries.

We have already begun investigating strategies for incremental maintenance of window states and we soon expect more concrete algorithms that can boost performance and approximation accuracy. Next, we also plan to conduct a comprehensive empirical study against real and synthetic datasets in order to attest scalability and robustness of the proposed concepts.

References

1. Bertino, E., Camossi, E., Bertolotto, M.: Multi-granular Spatio-temporal Object Models: Concepts and Research Directions. In: Norrie, M.C., Grossniklaus, M. (eds.) ICOODB 2009. LNCS, vol. 5936, pp. 132–148. Springer, Heidelberg (2010)
2. Bettini, C., Dyreson, C.E., Evans, W.S., Snodgrass, R.T., Sean Wang, X.: A Glossary of Time Granularity Concepts. In: Etzion, O., Jajodia, S., Sripada, S. (eds.) Dagstuhl Seminar 1997. LNCS, vol. 1399, pp. 406–413. Springer, Heidelberg (1998)
3. Cao, H., Wolfson, O., Trajcevski, G.: Spatio-temporal Data Reduction with Deterministic Error Bounds. VLDB Journal 15(3), 211–228 (2006)
4. Egenhofer, M.J., Franzosa, R.D.: Point-Set Topological Spatial Relations. IJGIS 5(2), 161–174 (1991)
5. Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: A Data Model and Data Structures for Moving Objects Databases. In: ACM SIGMOD, pp. 319–330 (2000)
6. Gedik, B., Liu, L., Wu, K., Yu, P.S.: Lira: Lightweight, Region-aware Load Shedding in Mobile CQ Systems. In: ICDE, pp. 286–295 (2007)
7. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A Foundation for Representing and Querying Moving Objects. ACM TODS 25(1), 1–42 (2000)
8. Güting, R.H., Behr, T., Düntgen, C.: SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations. IEEE Data Engineering Bulletin 33(2), 56–63 (2010)
9. Meratnia, N., de By, R.A.: Spatiotemporal Compression Techniques for Moving Point Objects. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 765–782. Springer, Heidelberg (2004)
10. Parent, C., Spaccapietra, S., Zimányi, E.: The MurMur Project: Modeling and Querying Multi-Representation Spatio-Temporal Databases. Information Systems 31(8), 733–769 (2006)
11. Patroumpas, K., Sellis, T.: Multi-granular Time-based Sliding Windows over Data Streams. In: TIME, pp. 146–153 (2010)
12. Patroumpas, K., Sellis, T.: Maintaining consistent Results of Continuous Queries under Diverse Window Specifications. Information Systems 36(1), 42–61 (2011)
13. Potamias, M., Patroumpas, K., Sellis, T.K.: Online Amnesic Summarization of Streaming Locations. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 148–166. Springer, Heidelberg (2007)
14. Stonebraker, M., Çetintemel, U., Zdonik, S.: The 8 Requirements of Real-Time Stream Processing. ACM SIGMOD Record 34(4), 42–47 (2005)

A Geographic Standards Based Metamodel to Formalize Spatio-temporal Knowledge in Remote Sensing Applications

Christelle Pierkot

IRD UMR Espace-Dev,
500 rue JF Breton,
34093 Montpellier, France
lastname.firstname@ird.fr

Abstract. Technological tools allow the generation of large volumes of data. For example satellite images aid in the study of spatio-temporal phenomena in a range of several disciplines, e.g. environment and health. Thus, remote-sensing experts must handle various and complex image sets for their interpretations. Additionally, the GIS community has heavily worked on describing spatio-temporal features, and standard specifications nowadays provide design foundations for GIS software and spatial databases. We argue that this spatio-temporal knowledge and expertise would provide invaluable support for the field of image interpretation. As a result, we propose a high level conceptual framework, based on existing and standardized approaches, offering enough modularity and adaptability for representing the various dimensions of spatio-temporal knowledge.

Keywords: Spatio-temporality, geographic standards, remote-sensing interpretation

1 Introduction and Objectives

Technological tools allow generating huge volumes of data, like satellite images, helping the study of spatio-temporal phenomena in various research fields, such as environment, health, or ecology. The GIS community, whose aim is to propose tools to exploit these data, has been very active in modeling spatio-temporal knowledge for many years [1], [4], [8], [9], [15]. Some of the work resulted in standard specifications and recommendations from OGC and ISO organizations [12], [14], and provide design foundations to both GIS software and spatial databases. However, at this point remote sensing tools used by the experts (e.g. Ecognition, OTB) poorly integrate these spatio-temporal aspects, leading to a lack of unified formalization. Thus, it is necessary to propose solutions for taking the spatio-temporal knowledge in scientific applications using satellite images as decision support into account most efficiently.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 304–313, 2012.

We argue that the spatio-temporal expertise acquired by the GIS community would provide invaluable support to the field of image interpretation. As a result, we propose a high level conceptual framework offering enough modularity, adaptability and genericity to formalize the dimensions of spatio-temporal knowledge. Indeed, this metamodel gives a standardized semantic description of the spatio-temporal concepts that can be formalized into a framework ontology, used to design domain ontologies according to specific application contexts and objectives, e.g. urban planning or land cover mapping.

This paper is structured as follows. First, we introduce our metamodel based on standards and previous work and we detail each component individually (2). Then, we provide an application example where the metamodel was used to conceptualize expert knowledge in the field of Amazonian Biome land covering (3). Finally, we conclude and give some perspectives (4).

2 Spatio-temporal Metamodel

The proposed metamodel is intended to be used by people that handle satellite images to make interpretations to understand spatio-temporal phenomena. These analyses are made in various themes (e.g. land cover, health, biodiversity), by experts with varying skills (e.g. ecologist, mapmaker) and with distinct objectives (e.g. land cover mapping, phenomena tracking, health monitoring). Thus, the model needs to be easily understood, sufficiently expressive to satisfy all information met in the diverse themes and flexible regardless of the context and objectives. Additionally, to better interpret satellite images, experts need to combine image information (e.g. shape of the object is a polygon and spectral signature defines its nature as vegetation) and field knowledge (e.g. mangroves grow in salt water and are located between ocean and continent). However, (1) all knowledge must be formalized to be usable, and (2) matching between the *image and field viewpoints* is then necessary to exploit information (e.g. find that object in image viewpoint correspond to mangrove concept in field viewpoint). Thus, the metamodel must also be generic enough to represent characteristics of distinct viewpoints and sufficiently modular to make matching easier.

Spatio-temporal information exists in both image and field viewpoints (e.g. shape of a feature). Some progress has been made to model information with spatial and/or temporal dimensions, but either the approaches were not designed especially to track spatio-temporal phenomena [3], as they are dedicated for only one particular phenomenon [2], or they have been designed for other tools, such as GIS software or spatial databases [10], [15]. Anyway, at the best of our knowledge, no attempt has been made to conceptualize a high level spatio-temporal framework in the particular field of remote sensing interpretation. Otherwise, metamodels exist in ISO and OGC standards to describe spatio-temporal features (e.g. General Feature Metamodel from ISO191xx standards), but they are not well suitable for interpreting satellite images (e.g. matching both field and remote sensing experts viewpoints). However, some parts of these approaches can be reused to express some of the spatio-temporal phenomena that are also in the field of remote sensing, such as geometries or temporal events of a feature.

In the following we present our metamodel by focusing on the way we organize the information to consider the geographic standards and integrate major reference work [1], [8], [12], [14].

2.1 Metamodel Structure

A general view of our metamodel is presented in figure 1, where eight components have been identified to define spatio-temporal knowledge in the field of image interpretation. We organize them as UML packages in order to aggregate information semantically close and to ensure modularity.

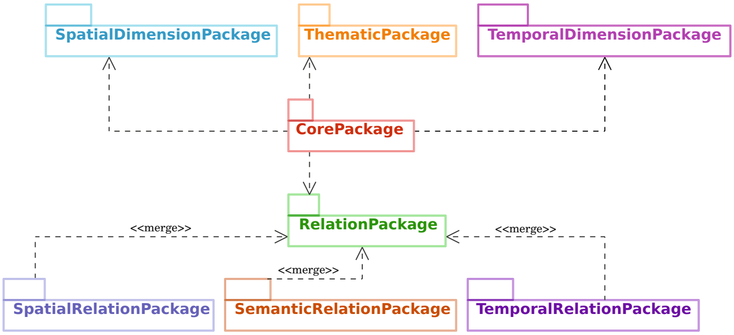


Fig. 1. Spatio-temporal packages

2.2 The Core Package

The *Core package* is the central element of the metamodel. It is used to characterize the geographical feature as a whole and to have a direct or indirect dependency on the other packages. It is a shared opinion that a geographic feature is an object, which represents an abstraction of a real world phenomenon with a local position from the earth [14], [12], [15]. However, the feature is defined with more or less complexity by normative organisms (only by geometry in [16], by one type and some geometric or thematic attributes in [14], and by some attributes, relationships and operations in [12]). By taking these approaches into account, we define a geographic feature as an *aggregation of spatial, temporal and thematic dimensions, with which different kinds of relations can be specified.*

The originality of this conceptualization relies on the following points: First, *thematic* is defined as a class and not as an attribute of the class feature. Due to this conceptualization, it is possible to take the different points of view associated with one geographic feature into account. For example, in field viewpoint, the *thematic class* describes relative domain properties of the concept (e.g. mangrove characteristics), while in image viewpoint, *thematic class* describes physical

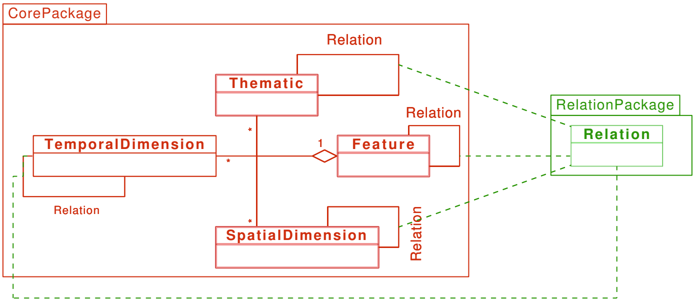


Fig. 2. Core package

properties of the object (e.g. IR spectral band values). Secondly, relationships are specified by each core class (i.e. feature, thematic and spatio-temporal dimensions) and not only on the geographic feature. Thus, according to the different point of view, we can explicitly specify which element is affected by the relation. For example, in a remote sensing image (image point of view), spatial relation between two features is generally defined by the geometry, which is a spatial dimension concept, while the feature itself will be used by the expert in the field viewpoint. Finally, differently to the other models, we have chosen to represent relations as an association class, which will be reified. The aim here is to add specific information such as the reference system used to define a projective spatial relation as properties.

2.3 Spatial Dimension

The *SpatialDimensionPackage* contains information about spatial references of the feature. It is directly linked with the core package by the *SpatialDimension* class.

Usually, the spatial dimension of a feature is defined by a *location* and a geometric *shape* [12], [15]. The feature position is provided by geographic or planar *coordinates* or by an approximation like a *bounding box*. However, whatever the representation mode, it is necessary to add the associated geodesic *reference system*. Since the points-set topological theory defined by [8], many attempts have been made to specify the geometry of a geographic feature, which have been included in the standards works (e.g. ISO19107, ISO 19125-1, OGC Features) [12], [14]. Thus, we add a class *NormalizedShape* to our metamodel to set the concepts defined in the OGC and ISO standards. However, our approach allows to describe the shape by concepts taken from the application domain through the class *OtherShape*. Indeed, in a satellite image, we can extract the feature shape concepts with some classes defined by remote sensing software (e.g. Ecognition or OrpheoToolBox). Finally, spatial referencing can be made by using geographic identifiers defined in ISO 19112 [12], and some databases with toponym concepts

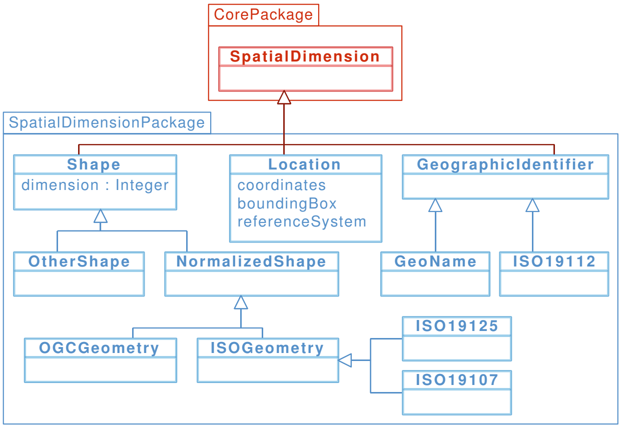


Fig. 3. Spatial Dimension package

can be used to transform the identifier in coordinates. It is useful to denominate a feature in the field point of view (e.g. expression ”Cayenne’s coastal mangrove” can be transform in a bounding box that delineates the affected area). Therefore, we propose to add a class name *GeographicIdentifier* to our conceptualization to take this information into account.

2.4 Temporal Dimension

The *TemporalDimensionPackage* includes concepts, which characterize time. It is directly linked to the core package by the *TemporalDimension* class.

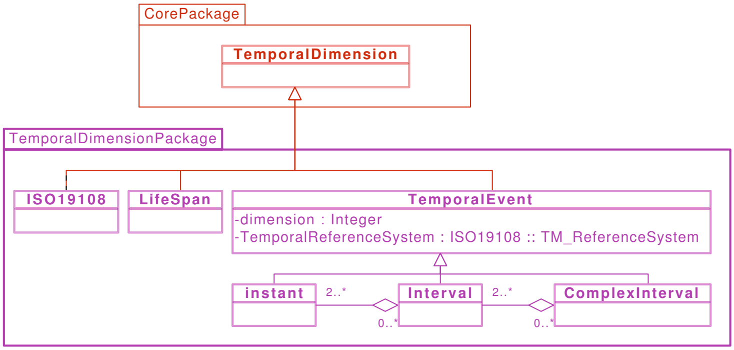


Fig. 4. Temporal Dimension package

In the literature, there are two ways for describing temporality: talking about time or modeling the change [4], [13], [15], [18]. ISO19108 standard [12] deals with time, which is considered as a dimension by analogy to what is carried out in ISO19107 for the spatial concepts. Based on these approaches we define three classes, each one could describe geographic features by the temporal dimension: *LifeSpan* class is dedicated to associate a terminological information with a geographic feature such as creation or evolution. *TemporalEvents* are defined by *instant* (e.g. acquisition date of the image), *interval* (e.g. flood period) or a *complex interval* composed by a set of disjointed intervals (e.g. seasons, if we consider several years). They allow representing numerical information relative to a geographic feature. We use the *TM_ReferenceSystem* from ISO 19108 to specify the reference system corresponding to each event [12]. Finally, we take *ISO19108* standard into account, by adding a specific class in our metamodel.

2.5 Thematic Dimension

The *Thematic package* aim is to describe the other nature of a feature, such as image characteristics or landscape properties. It is directly linked to the core by the *Thematic* class. It is defined by a set of concepts and relationships that are relevant for a domain of study, as for example the physical properties of an image (e.g. spectral band, texture) or the description of a landscape (e.g. Amazonian biome). Thus, the thematic dimension could be specified only when the domain is known, in the model derived from the metamodel.

2.6 Relations

The *Relation package* contains all the required concepts for describing a relationship between features. It is directly linked to the *CorePackage* by the *Relation* association class, which is specialized into three sub-classes in order to refine

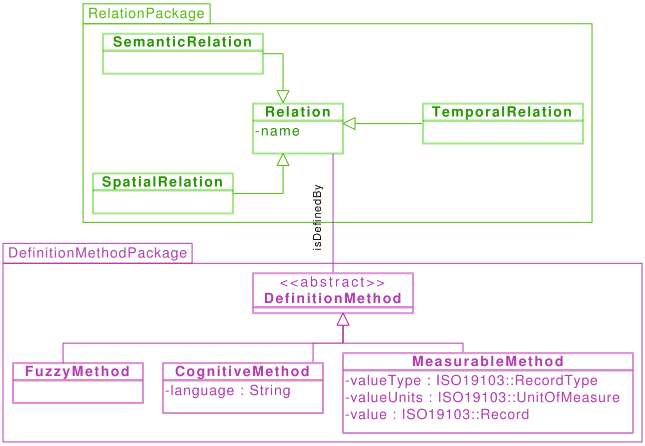


Fig. 5. Relation package and definition methods

it in terms of *Spatial, Temporal and Semantic relations*. Additionally, we provide three definition methods to specify each type of relations according to the taken viewpoint: (1) *MeasurableMethod* are methods that define relations with numerical values (measured or calculated), such as currently used in standards. (2) *CognitiveMethod* are methods used to define terms given by the expert to describe relations. And (3) *FuzzyMethod* are instantiations of relations defined by the two previous methods on a [0,1] interval.

Spatial Relations

The *SpatialRelation package* includes concepts to define spatial relations between features, such as "near" or "50m away".

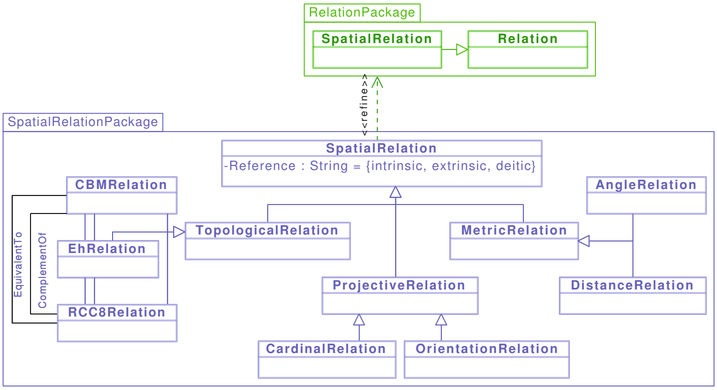


Fig. 6. Spatial Relation package

Many directions have been taken to define spatial relations [6], [7], [8], and are currently used in the standards [12], [14]. We use the types defined in [5] to specify three classes of spatial relations i.e. *topological, projective and metric*. *Metric relations* are of distances or angles [9]. They can be defined by measurable methods (e.g. the town is located 5km away from the beach), cognitive methods (e.g. forest is near river), or fuzzy methods.*Topological relations* are about connections between objects. These relationships are generally defined by measurable methods (e.g. via the DE9IM matrix [8]), but can also be expressed by terminologically cognitive methods (e.g. next to, touches, within). Three approaches are regularly cited in the literature, namely, the *point set based model of nine intersection* by [8] (*EhRelation*), the *Logic based Model connection calculus regions* of [7] (*RCC8Relation*), and the *Calculus based model* of [6] (*CBMRelation*). We choose to explicitly define these three classes in our metamodel, because they are commonly used by several communities and they can be easily linked to each other [16]. *Projective relations* are described by space projections such as cardinal relationship (e.g. east of, north of) [9], or orientation relations of the objects against each other (e.g. left, down, front) [11]. Finally, we choose to represent reference systems used with the relation by an attribute, whose type is defined in [17] (i.e. intrinsic, extrinsic and deitic).

Temporal and Semantics Relations

The *TemporalRelation package* includes concepts to define temporal relationships between features, such as "before" or "four months ago". Seven temporal relations have been defined by Allen [1], and have been specified in the ISO19108 standards [12]. We take these relations in our metamodel into account, and we argue that temporal relations can exist on features, but also on their spatial (e.g. widening of a river bed during a flood) and/or thematic dimensions (e.g. evolution of culture types in a registered land). Finally, just like spatial relations, temporal relations can be defined by measurable methods (e.g. since 3 hours), cognitive methods (e.g. before, after) or fuzzy methods (e.g. about a year).

The *SemanticRelation package* includes all the others relations that can exist between features such as *part of*, *is a*, *grow*, ... As for thematic dimension, some of the most common semantic relations depend on the domain and cannot be explicitly specified in the metamodel (excepting *is a* and *part of* relations, which are generally used to define aggregation and specialization relationships). Class *semantic relation* therefore serves as an anchor to the package relationship that will only be used at the model level.

3 An Application Example

An application model was derived from this metamodel to specify information about the Amazonian Biome, according to the field point of view. We first focused our efforts on the description of the concepts relating to the domain of study. At

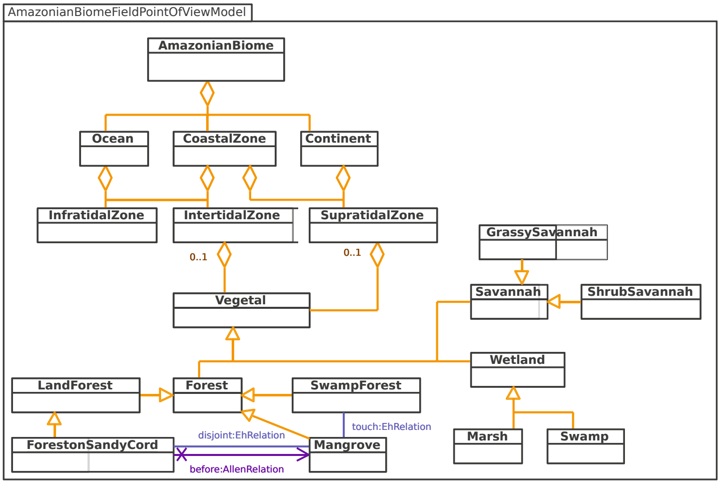


Fig. 7. Amazonian biome ontology in the field point of view

this step, we only use the two semantic relations *is a* and *part of*, to describe aggregation and specialization relationships. As for example, in this conceptualization, *Mangrove* which is the focussing concept of the study is defined as a sort of *Forest*, which is also a *Vegetal*. Then, this model was refined by adding spatial and temporal relations in accordance with the specifications given in the *Relation packages*. Thus, to describe topological relationships, experts must take advantage of concepts defined in the metamodel. If it is not adequate, they can propose new terms, stored as cognitive ones in the model. For example, to specify that spatial contact exists between the swamp forest and the mangrove, expert used the *Touch* topological relation from the *EhRelation* class. Further, to express that the formation of forests on sandy cords is a thousand years older than that of Mangrove, the expert uses the *Before* temporal relation from *Allen* class.

4 Conclusions and Perspectives

In this paper, we present a conceptual metamodel, based on normalized approaches, that can be used as a framework in the remote sensing domain to formalize spatio-temporal knowledge. Its aim is to support the image interpretation by experts in various research fields (e.g. ecology, health and environment) and according to the associated point of view (e.g. field or image reality). Thus, this metamodel was designed in a modular way, so that each package can be specified individually, facilitating the conceptual work of experts. Therefore, the experts focus only on the formalization of their expertise domain and integration becomes easier as a result. As experiment, thematic experts use the metamodel to conceptualize the Amazonian biome in the field point of view. This first application has demonstrated the ease of use of the metamodel to describe spatio-temporal knowledge in a particular viewpoint. The next step will be to use this metamodel in the image point of view for conceptualizing the objects of the domain (e.g. vegetated segment) in a modular way. Then, we will formalize all the knowledge in OWL ontologies and we will match both viewpoints in order to define consistent links. Finally, we will use description logics and reasoning to support image interpretation for the purpose of land cover classification.

Acknowledgements. The research leading to these results has received funding from the PO FEDER GUYANE 2007-2013 program under the frame of the CARTAM-SAT project.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM 26, 832–843 (1983)
2. Antoniadis, A., Helbert, C., Prieur, C., Viry, L.: Spatio-temporal metamodeling for West African monsoon. Environmetrics 23(1), 24–36 (2012)
3. Berger, S., Grossmann, G., Stumptner, M., Schrefl, M.: Metamodel-Based Information Integration at Industrial Scale. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010, Part II. LNCS, vol. 6395, pp. 153–167. Springer, Heidelberg (2010)

4. Claramunt, C., Thériault, M.: Managing time in gis: An event-oriented approach. In: Proceedings of the International Workshop on Temporal Databases: Recent Advances in Temporal Databases, pp. 23–42. Springer, London (1995)
5. Clementini, E.: A Conceptual Framework for Modelling Spatial Relations. Phd in computer sciences, INSA Lyon (2009)
6. Clementini, E., Di Felice, P.: A model for representing topological relationships between complex geometric features in spatial databases, vol. 90, pp. 121–136 (1996)
7. Cohn, A.G., Bennett, B., Gooday, J., Gotts, N.M.: Qualitative spatial representation and reasoning with the region connection calculus, vol. 1, pp. 275–316. Kluwer Academic Publishers, Hingham (1997)
8. Egenhofer, M.: A Formal Definition of Binary Topological Relationships. In: Litwin, W., Schek, H.-J. (eds.) FODO 1989. LNCS, vol. 367, pp. 457–472. Springer, Heidelberg (1989)
9. Frank, A.U., Campari, I., Formentini, U. (eds.): GIS 1992. LNCS, vol. 639. Springer, Heidelberg (1992)
10. Goodchild, M.F., Yuan, M., Cova, T.J.: Towards a general theory of geographic representation in gis. International Journal of Geographical Information Science 21(3), 239–260 (2007)
11. Hernández, D.: Qualitative Representation of Spatial Knowledge. LNCS, vol. 804. Springer, Heidelberg (1994)
12. ISO/TC211. Iso geographic information/geomatics technical committees 211 (1994)
13. Lardon, S., Libourel, T., Cheylan, J.P.: Concevoir la dynamique des entités spatio-temporelles. In: Représentation de l'Espace et du Temps dans les SIG, pp. 45–65 (1999)
14. OGC/TC. Opengis abstract specification (1999)
15. Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual modeling for traditional and spatio-temporal applications - the MADS approach. Springer (2006)
16. Perry, M., Herring, J.: Ogc geosparql, a geographic query language for rdf data. Technical report, Open Geospatial Consortium (2011); OGC candidate standard
17. Retz-Schmidt, G.: Various Views on Spatial Prepositions. AI Magazine 9(2), 95–105 (1988)
18. Worboys, M.: Event-oriented approaches to geographic phenomena. International Journal of Geographical Information Science 19, 1–28 (2005)

A Fuzzy Spatio-temporal-Based Approach for Activity Recognition

Jean-Marie Le Yaouanc and Jean-Philippe Poli

CEA, LIST, 91191 Gif-sur-Yvette CEDEX, France
firstname.lastname@cea.fr

Abstract. Over the last decade, there has been a significant deployment of systems dedicated to surveillance. These systems make use of real-time sensors that generate continuous streams of data. Despite their success in many cases, the increased number of sensors leads to a cognitive overload for the operator in charge of their analysis. However, the context and the application requires an ability to react in real-time. The research presented in this paper introduces a spatio-temporal-based approach the objective of which is to provide a qualitative interpretation of the behavior of an entity (*e.g.*, a human or vehicle). The process is formally supported by a fuzzy logic-based approach, and designed in order to be as generic as possible.

Keywords: Spatio-temporal data modeling, Automatic activity recognition, Semantic trajectories, Fuzzy logic.

1 Introduction

Surveillance is of growing interest because of the importance of safety and security issues. When integrated with Geographical Information Systems (GIS), systems dedicated to surveillance combine spatial features with the information provided by real-time sensors to provide a support for the management of entities (*e.g.*, humans or vehicles). Supervision of mobile entities has a wide range of potential applications, such as the security and safety of critical buildings (*e.g.* stadiums, airports), or the traffic surveillance in cities. However, the increase use of sensors leads to a saturation for the human operator in charge of the data analysis. Consequently, it would be desirable to develop systems that assist humans in supervising spatial scenes, *i.e.* systems that automatically analyze data streams, detect suspicious events, and advise an operator to check a particular screen.

Automatic activity recognition is a process the objective of which is to interpret the behavior of entities in order to generate a description of the detected events or to raise an alarm. The capture of information associated to these entities is operated by sensors such as video cameras that collect images of a specific environment, or geo-positioning systems that record geographical positions. Using time intervals and logical formalisms, previous approaches have obtained successful results in detecting high level activities [1]. Formal rules have been

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 314–323, 2012.

defined for detecting activities involving interactions amongst people or recognize unusual behaviors of individuals [2,3]. More recently, an expert system was used to combine facts detected by a low-level framework, and inference rules previously defined by an expert [4]. Petri nets have also been considered as a well-adapted representation and recognition support [5]. F. Bremond rather suggests the use of finite state machines, where states represent sub-activities, and transitions, the events. An activity is recognized if its final state is reached [6]. Hereafter, we focus on the automatic recognition of activities from the interpretation of trajectories. Spatio-temporal configurations between two mobile entities can be detected by analyzing their relative distances and speeds [7,8]. B. Gottfried defines a spatio-temporal model based on the analysis of the evolution of relative directions between two mobile entities [9]. Other models are specifically designed for spatial databases, and particular operators are defined that optimize the implementation of complex spatio-temporal queries [10,11]. However, these approaches have limitations in handling uncertainties and variations since they identify activities only when their spatial and temporal relationships are strictly satisfied, ignoring the variations. As a matter of fact, the execution of an event is usually dependent of the context and the intrinsic characteristics of the entity.

The research presented in this paper concerns the real-time semantic interpretation of the behavior of a mobile entity observed by sensors. Real-time sensors generate a huge amount of quantitative data. However, these data do not completely reflect the way a human perceives and describes an environment since he preferably stores and processes qualitative information. As a consequence, we provide a semantic model suitable with cognition, but also appropriated for the processing of spatio-temporal data. The model analyses the quantitative data recorded by sensors and evaluates behaviors involving entities (*i.e.* humans or vehicles). Since it is designed to consider uncertainties of the activities' structures, the qualitative interpretation is supported by a fuzzy-based approach that provides a fuzzy interpretation of the spatial and temporal dimensions.

The reminder of the paper is organized as follows. Section 2 briefly introduces basic principles on fuzzy logic. Section 3 provides a conceptual representation of an activity and models fuzzy spatio-temporal relations. Finally, Section 4 draws the conclusions and outlines further work.

2 Basic Principles on Fuzzy Logic

Fuzzy logic was designed to allow systems to mimic the way humans think. Fuzzy logic is based upon the fuzzy set theory that is a formal mathematical theory dedicated to the representation of uncertainty [12]. The approach is particularly relevant when dealing with real world systems that interact with humans, since humans mainly manipulate qualitative information. Hereafter, we briefly outline how fuzzy logic extends classical logic. Let us denote X a universe of discourse, a fuzzy subset $A \subset X$ is characterized by its *membership function* μ_A,

$$\mu_A : X \to [0, 1]$$

For each $x \in X$, the value $\mu_A(x)$ is interpreted as the degree of membership of x in the fuzzy set A, or, equivalently, as the *truth value* of the proposition "x is an element of A". In order to generalize the set theoretical operations intersections and unions, triangular norms (t-norms) and conorms (t-conorms) were defined. Although there are many ways to define t-norms and t-conorms, only few are used in applications. One of the most used t-norm, together with its dual t-conorm is the one defined by L. Zadeh: $x \wedge y = \min(x, y), x \vee y = \max(x, y)$.

3 Modeling Approach

3.1 Conceptual Modeling of an Activity

We model an activity by a Situation Graph Tree (SGT) [13], the objective of which is to facilitate the understanding of the structures that emerge from the description of an activity. SGT are hierarchical trees that characterize the behavior of entities in terms of situations they can be in. Such a graph illustrates the combination of elementary units that model a particular situation with hierarchical (*e.g.*, a situation composed of several sub-situations), temporal (*e.g.*, a situation that occurs before or while another one) and semantic relations. As a matter of fact, the semantics related to an entity at a given time is contained in an elementary unit, which constitutes the basic component of a SGT. Elementary units that represent different temporal episodes of the same situation are enclosed by the situation graph. We characterize an elementary unit as a semantic function that qualitatively evaluates a situation or an action. An elementary unit relates an entity (that may be dynamic or static) with a spatial object, *i.e.* a landmark or a form of the environment. Landmarks are salient objects that structure a cognitive representation of an environment [14]. They constitute key-references for the conceptualization and the description of an environment, and consequently play a prominent role for describing a spatial situation or characterizing the movement of an entity in an environment. The principles of the modeling approach being introduced, we hereafter present the formal representation of an elementary unit. Let $\mathbb{G}$ be the set of SGT, $\mathbb{U}$ the set of elementary units composing a SGT, $\mathbb{E}$ the set of mobile entities, $\mathbb{R}$ the set of spatio-temporal relations and $\mathbb{O}$ the set of simple spatial objects, *i.e.*, landmarks or forms that structure an environment. A situational graph tree $\mathcal{G} \in \mathbb{G}$ is an ordered set of elementary units $u_i \in \mathbb{U}$, *i.e.*, $\mathcal{G} = [u_1, \ldots, u_n]$ where $n \geq 1$. An elementary unit u_i is a triplet such as $u_i = [e_j, r_k, o_l]$ with $e_j \in \mathbb{E}, r_k \in \mathbb{R}$ and $o_l \in \mathbb{O}$.

3.2 Modeling of Spatio-temporal Relations

In the following subsections, the modeling of spatio-temporal relations that characterize the activity of a mobile entity in an environment is developed. In order to consider the uncertainties of the activities' structures, the semantic interpretation of the entity's trajectory is supported by a fuzzy-based approach. It is designed to be as generic as possible, and considers objects with rather *bona fide* or *fiat* boundaries. The former are objects with physical discontinuities (*e.g.* a mountain or a valley), the latter gets boundaries induced through human demarcation (*e.g.* a building or an administrative region) [15]. The development of formal models of topological relations has received much attention in the literature of GIS, computer vision and image understanding [16]. In recent years, significant achievement have been made on the development of formal models of topological relations between spatial objects with indeterminate boundaries. C. Hudelot and I. Bloch defined spatial relations such as the adjacency and inclusion, but also directional relations between fuzzy image regions [17]. Among the GIS community, E. Clementini and P. Di Felice [18], and A. Cohn and N. Gotts [19] developed models of topological relations between fuzzy regions, *e.g. Disjoint*, *Meet* or *Inside*.

Temporal representation and reasoning is also an important facet in the design of a fuzzy spatio-temporal approach. As a matter of fact, when the time span of an activity is imprecise, it can be represented by a fuzzy time interval. J. F. Allen defined a set of 13 qualitative relations, *e.g. Before* and *After*, that may hold between two intervals [20], and his work was recently extended to a more general formalism that can handle precise as well as imprecise relationships between crisp and fuzzy intervals [21]. P. Cariñena provides a complementary approach and models the temporal relations *Occurrence* and *Persistence* between an event and a fuzzy temporal interval [22].

Let $\mathbb{I}$ be the set of temporal intervals, $\mathbb{T}$ the set of instants, $\mathbb{O}$ the set of simple spatial objects, $\mathbb{O}_1 \subset \mathbb{O}$ the set of simple closed regions, $\mathbb{O}_2 \subset \mathbb{O}$ the set of simple opened regions, $\mathbb{P}$ the set of fuzzy propositions and $\mathbb{F}$ the set of fuzzy membership functions. Let $p \in \mathbb{P}$ be a fuzzy proposition, and $\mu(p,t)$ the value of p at a given moment t. Let $I \in \mathbb{I}$ be a temporal interval and t a given moment. We denote $I^* = I \setminus \{t\}$. Let $Mean$ be the function that computes the mean of a set of fuzzy values along a given interval I:

$$\begin{array}{lll} Mean: & \mathcal{F} \times \mathbb{I} & \to \mathcal{F}, \text{ with } \mathcal{F} \text{ the set of fuzzy values} \\ & (\mu(p,t), I) & \mapsto \frac{\sum\limits_{t \in I} \mu(p,t_i)}{Card(t)}, \text{ where } Card() \text{ is the cardinality operator.} \end{array}$$

Relation *IsMoving*. The spatio-temporal relation *IsMoving* characterizes the moving of a mobile entity in a non-constraint space [1]. Its evaluation takes into account the positions of the considered entity during a past time interval. It is based on the assumption that the value at time t_i may not only be based on the last moving between t_{i-1} and t_i, but on their recording in the past. Consequently, if the entity e is not moving between times t_{i-1} and t_i, the value of the relation *IsMoving* is pondered by its previous moving during a given past time interval. In other words, if a pedestrian stops walking at time t because he is looking for his keys, the value of the relation *IsMoving* will decrease in time if he stops during a significant time. More formally, let $I_1 \in \mathbb{I}$, $p_1 \in \mathbb{P}$ be the fuzzy

[1] The relation *IsMoving* relates an entity to the studied environment. To facilitate the reading, this object is not clearly mentioned.

proposition "the distance travelled by e is not zero". The fuzzy proposition p_1 is correlated to the moving distance of e between instants t_{i-1} and t_i. Figure 1 illustrates a possible representation of f_1, the fuzzy membership of p_1. Let *IsMoving()* be the function that models the moving of an entity,

$$\begin{aligned} IsMoving : \mathbb{E} \times \mathbb{F} \times \mathbb{I} \times \mathbb{T} &\to \mathcal{F}, \text{ with } \mathcal{F} \text{ the set of fuzzy values} \\ (e, f_1, I_1, t) &\mapsto \mu(p_1, t) \vee Mean(\mu(p_1, t), I_1^*). \end{aligned}$$

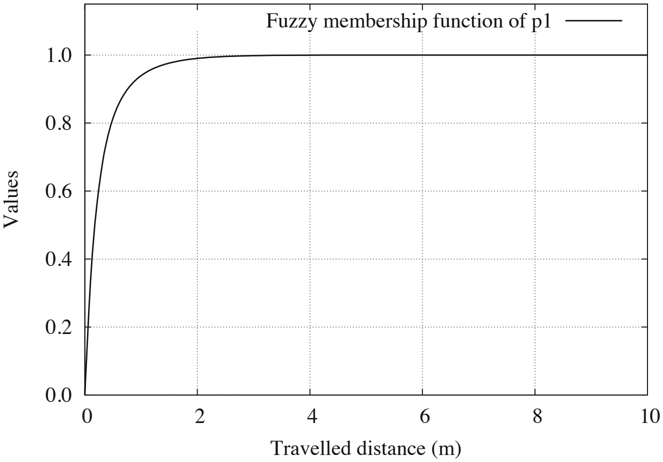


Fig. 1. Fuzzy membership function f_1

Relation *IsComingCloseTo*. The spatio-temporal relation *IsComingCloseTo* characterizes the approach of a spatial object $o \in \mathbb{O}$ by a mobile entity $e \in \mathbb{E}$ in a non-constraint space. For instance, this relation may be useful for characterizing a boat that is coming close to a navigational buoy. Intuitively, the closer the entity to o in a way that minimizes the distance to reach the object, the higher the fuzzy value of *IsComingCloseTo* is. Two spatial configurations are defined, with respect to the geometry of o.

We study the approach of an object modelled as a point or an open polyline. The geometry of the given object is provided by a vector geo-database that covers the studied environment. Let $P(t_i)$ be the position of e at time t_i, $N(t_i)$ the point that minimizes the distance between e and o at time t_i, and $\overrightarrow{P(t_i)Q(t_i)}$ the vector that identifies the direction of e at time t_i (Fig. 2). The evaluation of the relation takes into account:

- the evolution of the location of e between times t_i and t_{i-1}.
- the orientation $\alpha(t_i) = (\overrightarrow{P(t_i)N(t_i)}, \overrightarrow{P(t_i)Q(t_i)})$ of e at time t_i. Thus, the more $\cos(\alpha(t_i))$ tends to 1, the higher the value of *IsComingCloseTo*.
- the recording of orientations $\alpha(t_i)$ during a temporal interval $I^* \in \mathbb{I}$. Thus, the more $Mean(\max(\cos(\alpha(t_i)), 0), I^*)$ tends to 1, the higher the value of *IsComingCloseTo*.

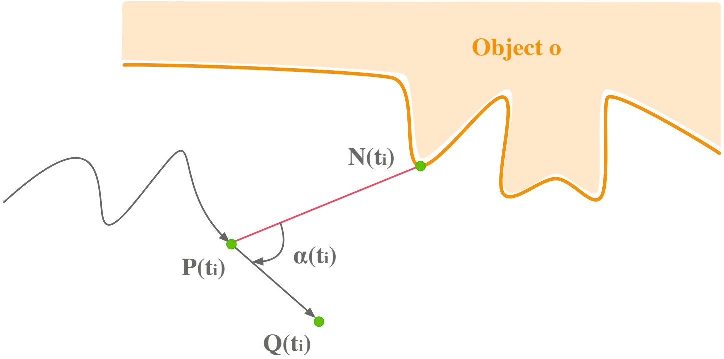


Fig. 2. Approach of an object modelled as a point or an open polyline

Let $I_1 \in \mathbb{I}$, $p_2 \in \mathbb{P}$ be the fuzzy proposition "the value of $\max(\cos(\max(\alpha(t_i), 0)), 0)$ tends to 1" and $f_2 \in \mathbb{F}$ its fuzzy membership. Figure 3 illustrates a possible representation of f_2, the fuzzy membership of p_2.

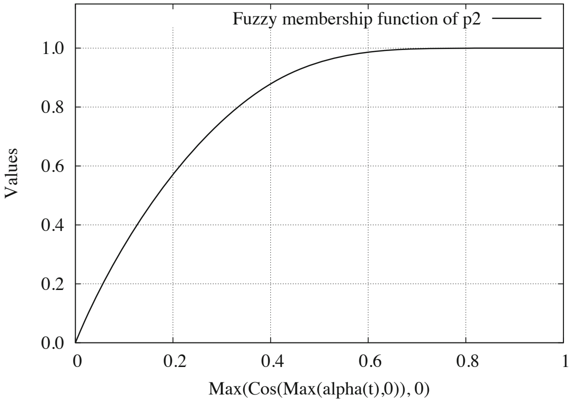


Fig. 3. Fuzzy membership function f_2

Let us denote *IsComingCloseTo()* the function that models the approach of an entity,

$$\begin{aligned} IsComingCloseTo : \mathbb{E} \times \mathbb{O}_2 \times \mathbb{F}^2 \times \mathbb{I} \times \mathbb{T} &\rightarrow \mathcal{F}, \text{ with } \mathcal{F} \text{ the set of fuzzy values} \\ (e, o, f_1, f_2, I_1, t) &\mapsto IsMoving(e, f_1, I_1, t) \\ &\quad \wedge(\mu(p_2, t) \vee Mean(\mu(p_2, t), I_1^*)) \end{aligned}$$

We consider now the approach of an object modelled as a closed region. Let $P(t_i)$ be the position of e at time t_i, Δ_1 and Δ_2 the exterior tangents of object $o \in \mathbb{O}_1$ that pass through the point $P(t_i)$, $P_{\Delta_1}(t_i)$ and $P_{\Delta_2}(t_i)$ the tangent points to o and the tangents, $(P(t_i)M(t_i))$ the median line that bisects the angle $(\overrightarrow{P(t_i)P_{\Delta_1}(t_i)}, \overrightarrow{P(t_i)P_{\Delta_2}(t_i)})$, and $\overrightarrow{P(t_i)Q(t_i)}$ the vector that identifies the direction of e at time t_i. Let us denote $\beta(t_i) = (\overrightarrow{P(t)M(t_i)}, \overrightarrow{P(t_i)P_{\Delta_1}(t_i)})$ and $\alpha(t_i) = (\overrightarrow{P(t_i)M(t_i)}, \overrightarrow{P(t_i)Q(t_i)})$ (Fig. 4). Intuitively, the lesser the value of angle $\alpha(t_i)$, the higher the value of the relation. However, as soon as the direction of e is inside in the directional cone $(\overrightarrow{P(t_i)P_{\Delta_2}(t_i)}, \overrightarrow{P(t_i)P_{\Delta_1}(t_i)})$ (represented in grey in Figure 4), the relation gets the maximal fuzzy value, *i.e.*, 1.

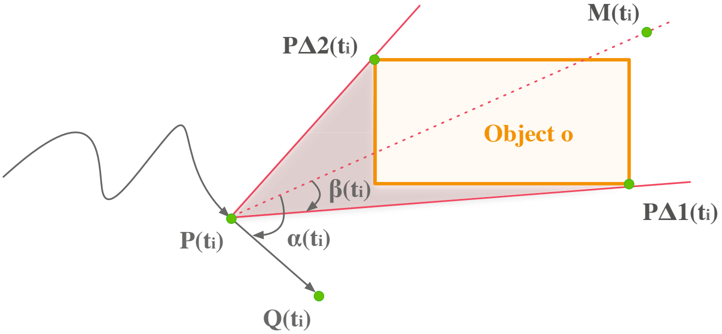


Fig. 4. Approach of an object modelled as a closed region

The evaluation of the relation takes into account:

- the location of e relatively to o
- the evolution of the location of e between times t_i and t_{i-1}.
- the orientation $\alpha(t_i)$ of entity e at time t_i. Thus, the more $\cos(\max(\alpha(t_i) - \beta(t_i), 0))$ tends to 1, the higher the value of *IsComingCloseTo* is.
- the recording of orientations $\alpha(t_i)$ during a temporal interval $I^* \in \mathbb{I}$. Thus, the more $Mean(\max(\cos(\max(\alpha(t_i) - \beta(t_i), 0)), 0), I^*)$ tends to 1, the higher the value of *IsComingCloseTo*.

Let $I_1 \in \mathbb{I}$, $p_3 \in \mathbb{P}$ the fuzzy proposition "the value of $\max(\cos(\max(\alpha(t_i) - \beta(t_i), 0)), 0)$ tends to 1" and $f_3 \in \mathbb{F}$ its fuzzy membership. Let us denote *IsComingCloseTo()* the function that models the approach of an object,

$$\begin{aligned} IsComingCloseTo : \mathbb{E} \times \mathbb{O}_1 \times \mathbb{F}^2 \times \mathbb{I} \times \mathbb{T} &\rightarrow \mathcal{F}, \text{ with } \mathcal{F} \text{ the set of fuzzy values} \\ (e, o, f_1, f_3, I_1, t) &\mapsto \mu(Persistence(Disjoint(e, o), I_1)) \\ &\quad \wedge IsMoving(e, f_1, I_1, t) \\ &\quad \wedge (\mu(p_3, t) \vee Mean(\mu(p_3, t), I_1^*)) \end{aligned}$$

Relation *IsGoingAway.* The spatio-temporal relation *IsGoingAway* characterizes the moving away of a mobile entity $e \in \mathbb{E}$ from a spatial object $o \in \mathbb{O}$ in a non-constraint space. For instance, this relation may be useful for characterizing a boat that moves away for the coast. Intuitively, the more the entity moves away from o in a way that maximize the distance to reach the object, the higher the fuzzy value of *IsGoingAway*. Two spatial configurations that are similar to the case illustrated in Section 3.2 are defined, with respect to the geometry of o.

We study the approach of an object modelled as a point or an open polyline. The evaluation of the relation takes into account:

- the evolution of the location of e between times t_i and t_{i-1}.
- the orientation $\alpha(t_i)$ of entity e at time t_i. Thus, the more $\cos(\alpha(t_i))$ tends to -1, the higher the value of *IsGoingAway*.
- the recording of orientations $\alpha(t_i)$ during a temporal interval $I^* \in \mathbb{I}$. Thus, the more $Mean(\min(\cos(\alpha(t_i)), 0), I^*)$ tends to -1, the higher the value of *IsGoingAway*.

Let $I_1 \in \mathbb{I}$, $p_4 \in \mathbb{P}$ the fuzzy proposition "the value of $\min(\cos(\alpha(t_i)), 0)$ tends to -1" and $f_4 \in \mathbb{F}$ its fuzzy membership. Let us denote *IsGoingAway()* the function that models the moving away from an object,

$$\begin{aligned} IsGoingAway : \mathbb{E} \times \mathbb{O}_2 \times \mathbb{F}^2 \times \mathbb{I} \times \mathbb{T} &\rightarrow \mathcal{F}, \text{ with } \mathcal{F} \text{ the set of fuzzy values} \\ (e, o, f_1, f_4, I_1, t) &\mapsto IsMoving(e, f_1, I_1, t) \\ &\quad \wedge(\mu(p_4, t) \vee Mean(\mu(p_4, t), I_1^*)) \end{aligned}$$

Hereafter, we study the approach of an object modelled as a closed region. Figure 4 illustrates the considered spatial configuration. The evaluation of the relation takes into account:

- the location of e relatively to o
- the evolution of the location of e between times t_i and t_{i-1}.
- the orientation $\alpha(t_i)$ of entity e at time t_i. Thus, the more $\cos(\max(\alpha(t_i) - \beta(t_i), 0))$ tends to -1, the higher the value of *IsGoingAway*.
- the recording of orientations $\alpha(t_i)$ during a temporal interval $I^* \in \mathbb{I}$. Thus, the more $Mean(\min(\cos(\max(0, \alpha(t_i) - \beta(t_i))), 0), I^*)$ tends to -1, the higher the value of *IsGoingAway*.

Let $I_1 \in \mathbb{I}$, $p_5 \in \mathbb{P}$ the fuzzy proposition "the value of $\min(\cos(\max(0, \alpha(t_i) - \beta(t_i))), 0)$ tends to -1" and $f_5 \in \mathbb{F}$ its fuzzy membership. Let us denote *IsGoingAway()* the function that models the moving away from an object,

$$\begin{aligned} IsGoingAway : \mathbb{E} \times \mathbb{O}_1 \times \mathbb{F}^2 \times \mathbb{I} \times \mathbb{T} &\rightarrow \mathcal{F}, \text{ with } \mathcal{F} \text{ the set of fuzzy values} \\ (e, o, f_1, f_4, I_1, t) &\mapsto \mu(Persistence(Disjoint(e, o), I_1)) \\ &\quad \wedge IsMoving(e, f_1, I_1, t) \\ &\quad \wedge(\mu(p_5, t) \vee Mean(\mu(p_5, t), I_1^*)) \end{aligned}$$

Relation *IsGoingAlong.* The spatio-temporal relation *IsGoingAlong* characterizes the action of an entity $e \in \mathbb{E}$ that goes along a spatial object $o \in \mathbb{O}$ in a non-constraint space. For instance, this relation may be useful for characterizing a boat that sails along the coast. Intuitively, the more the entity significantly moves close to o, the higher the fuzzy value of *IsGoingAlong*. The evaluation of the relation takes into account:

- the evolution of the location of e between times t_i and t_{i-1}.
- the proximity of e to o during a significant time span.

Let $I_1 \in \mathbb{I}$, $p_6 \in \mathbb{P}$ the fuzzy proposition "the entity e is near the object o" and $f_6 \in \mathbb{F}$ its fuzzy membership. Let us denote *IsGoingAlong()* the function that models the action of going along an object,

$$\begin{aligned} IsGoingAlong : \mathbb{E} \times \mathbb{O} \times \mathbb{F}^2 \times \mathbb{I} \times \mathbb{T} &\rightarrow \mathcal{F}, \text{ with } \mathcal{F} \text{ the set of fuzzy values} \\ (e, o, f_1, f_6, I_1, t) &\mapsto \mu(Persistence(Disjoint(e, o), I_1)) \\ &\quad \wedge IsMoving(e, f_1, I_1, t) \\ &\quad \wedge \mu(Persistence(p_6, I_1), t) \end{aligned}$$

4 Conclusion

Current systems dedicated to automatic activity recognition do not consider the spatial and temporal uncertainties, and identify particular activities only when spatial and temporal relationships are strictly satisfied. However, the context and the environment may influence the behavior of a mobile entity. The research presented in this paper introduces an approach for qualifying the activities of a mobile entity in real time. It analyses the trajectory of mobile entities recorded by sensors and qualitatively evaluates their behavior. It is supported by a fuzzy-based approach that both provides a fuzzy interpretation of the spatial and temporal dimensions. We have designed four spatio-temporal relations that relate an entity to an object of the environment that may get *bona fide* or *fiat* boundaries. The approach is currently being implemented and evaluated. The spatial extension of our prototype is based upon *DotSpatial*, *i.e.* a .Net library that favors the integration of geographic data and spatial analysis.

Although experienced for elementary activities, the semantic approach may be applied to high level activities. Such a work may be assessed with the use of a fuzzy expert system. Further theoretical work concerns an extension of the ontological background of the approach and the development of complementary spatio-temporal relations, *e.g.*, *IsGoingThrough*, *IsEntering*, *IsGoingOut* and *IsFollowingARoute*.

Acknowledgements. This work was part of a collaboration between the CEA-LIST and EGIDIUM Technologies.

References

1. Allen, J.F., Ferguson, G.: Actions and events in interval temporal logic. Journal of Logic and Computation 4(5), 531–579 (2010)
2. Shet, V., Harwood, D., Davis, L.: VidMAP: Video monitoring of activity with Prolog. In: IEEE International Conference on Advanced Video and Signal based Surveillance, Como, Italy, pp. 224–229. IEEE Computer Society (2005)
3. Geerinck, T., Enescu, V., Ravyse, I., Sahli, H.: Rule-based video interpretation framework: Application to automated surveillance. In: Proceedings of the 5th International Conference on Image and Graphics, pp. 341–348. IEEE Computer Society, Washington, DC (2009)
4. Krausz, B., Herpers, R.: Metrosurv: Detecting events in subway stations. Multimedia Tools and Applications 50(1), 123–147 (2010)

5. Ghanem, N., Dementhon, D., Doermann, D., Davis, L.: Representation and recognition of events in surveillance video using petri nets. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society, Washington, DC (2004)
6. Bremond, F., Medioni, G.: Scenario recognition in airborne video imagery. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Santa Barbara, CA, USA. IEEE Computer Society (1998)
7. Van de Weghe, N., Cohn, A., Maeyer, P., Witlox, F.: Representing moving objects in computer-based expert systems: the overtake event example. Expert Systems with Applications 29, 977–983 (2005)
8. Noyon, V., Claramunt, C., Devogele, T.: A relative representation of trajectories in geographical spaces. GeoInformatica 11(4), 479–496 (2007)
9. Gottfried, B.: Interpreting motion events of pairs of moving objects. GeoInformatica 15(2), 247–271 (2011)
10. Erwig, M.: Toward spatiotemporal patterns. Spatio-Temporal Databases 1, 29–54 (2004)
11. Hornsby, K.S., King, K.: Modeling motion relations for moving objects on road networks. GeoInformatica 12(4), 477–495 (2008)
12. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
13. Arens, M., Nagel, H.-H.: Behavioral Knowledge Representation for the Understanding and Creation of Video Sequences. In: Günter, A., Kruse, R., Neumann, B. (eds.) KI 2003. LNCS (LNAI), vol. 2821, pp. 149–163. Springer, Heidelberg (2003)
14. Lynch, K.: The Image of the City. The MIT Press, Boston (1960)
15. Smith, B., Varzi, A.: Fiat and Bona Fide Boundaries: Towards an Ontology of Spatially Extended Objects. In: Frank, A.U. (ed.) COSIT 1997. LNCS, vol. 1329, pp. 103–119. Springer, Heidelberg (1997)
16. Zhan, F.B.: Approximate analysis of binary topological relations between geographic regions with indeterminate boundaries. Soft Computing - A Fusion of Foundations, Methodologies and Applications 2, 28–34 (1998)
17. Hudelot, C., Atif, J., Bloch, I.: Fuzzy spatial relation ontology for image interpretation. Fuzzy Sets Systems 159(15), 1929–1951 (2008)
18. Clementini, E., Felice, P.D.: Approximate topological relations. International Journal of Approximate Reasoning 16(2), 173–204 (1997)
19. Cohn, A.G., Gotts, N.M.: The 'egg-yolk' representation of regions with indeterminate boundaries. In: Burrough, P., Frank, A.M. (eds.) Specialist Meeting on Spatial Objects with Undetermined Boundaries, pp. 171–187. Taylor & Francis (1997)
20. Allen, J.F.: Maintaining knowledge about temporal intervals. Communication of the ACM 26(11), 832–843 (1983)
21. Schockaert, S., Cock, M.D., Kerre, E.E.: Fuzzifying Allen's temporal interval relations. IEEE Transactions on Fuzzy Systems 16(2), 517–533 (2008)
22. Cariñena, P., Bugarin, A., Mucientes, M., Barro, S.: A language for expressing fuzzy temporal rules. Mathware & Soft Computing 7, 213–227 (2000)

Ninth International Workshop on Web Information Systems Modeling (WISM 2012)

Preface

The international workshop on Web Information Systems Modeling (WISM) aims to discuss the most recent developments in the field of model-driven design of Web Information Systems (WIS). This is the ninth edition of the workshop, after eight successful editions organized in Brussels (2011), Vancouver (2010), Amsterdam (2009), Barcelona (2008), Trondheim (2007), Luxembourg (2006), Sydney (2005), and Riga (2004).

The invited paper by Bielikova and Rastocny addresses the management of knowledge tags, i.e., metadata used to annotate information artifacts on the Web. It is proposed the usage of a repository based on a NoSQL database to store tags and a Map-Reduce processing of SPARQL queries to manage tags. By means of use cases, the authors show that such an approach is more efficient than a classical triple store for knowledge tags maintenance.

The first regular paper by Belk investigates the relationship between user cognitive styles and user navigation behaviour on the Web. The user cognitive features are determined by means of a psychometric survey, while the navigation features are measured in a non-intrusive way by monitoring user navigation behaviour. Several innovative metrics are proposed for computing the user navigation features. A cluster-based analysis of the user features allows the identification of the relationships between the user cognitive style and user navigation behaviour.

The second paper by De Virgilio and Dolfi aims to improve the user navigation to the desired Web pages by adapting the results presentation of a keyword-based search engine over Web page RDF data. These adaptations exploit the notion of centrality in an RDF graph by proposing to users Web pages that correspond to the "central" nodes in the corresponding RDF graph. This approach is positively evaluated for both effectiveness and efficiency against a state-of-the-art keyword-based search system for RDF graphs.

The third paper by Restrepo-Arango et al. proposes a software architecture for a financial application able to integrate and present on the Web information regarding stock prices, news, and social media on the market perception on various entities related to a user portfolio. The authors propose an innovative ranking function for news and the use of a NoSQL database to store the Web gathered information. The evaluation of the information retrieval task shows that the proposed approach has a good performance with respect to precision and recall.

The fourth paper by Hogenboom describes an ontology-based framework for extracting information from Web news items to be used in an algorithmic trading setup. The framework is based on a natural language processing pipeline able to extract financial events from news using a sequence of steps, among which we distinguish a word sense disambiguation procedure and a lexico-semantic pattern-based event extraction method. By associating impacts to financial events, the proposed framework is able to support a trading algorithm to make better decision on financial markets.

We do hope that the previous topics on modeling applications on the Web have triggered the reader's interest to have a closer look at the workshop proceedings. Last, we would also like to thank all the authors, reviewers, and participants for their input and support for the workshop.

October 2012

Flavius Frasincar
Geert-Jan Houben
Philippe Thiran

Lightweight Semantics over Web Information Systems Content Employing Knowledge Tags

Mária Bieliková and Karol Rástočný

Institute of Informatics and Software Engineering,
Faculty of Informatics and Information Technologies, Slovak University of Technology,
Ilkovičova 3, Bratislava, Slovakia
name.surname@stuba.sk

Abstract. A model of web information system content is crucial for its effective manipulation. We employ knowledge tags – metadata that describe an aspect of an information artifact for purpose of the modeling. Knowledge tags provide a lightweight semantics over the content, which serves web information systems also for sharing knowledge about the content and interconnections between information artifacts. Knowledge tags represent not only content based metadata, but also a collaboration metadata, e.g. aggregations of an implicit user feedback including interactions with the content. To allow this type of metadata we should provide means for knowledge tags repository providing flexible and fast access for effective reasoning. In this paper we address issues related to knowledge tags repository and its automatic maintenance. Main design issues are based on considering dynamic character of the web of information artifacts, which results in content changes in time that can invalidate knowledge tags. We realized the web-scale repository via the MongoDB database. Proposed repository stores knowledge tags in Open Annotation model and supports inference via distributed SPARQL query processing algorithm for MapReduce.

Keywords: lightweight semantics, knowledge tag, annotation, maintenance, MapReduce, SPARQL, distributed repository.

1 Introduction

Effective web content manipulation such as personalized search, recommendation or context aware navigation requires explicit representation of a content model. It obviously includes metadata on the content as an explicit representation of the content model. Moreover, interactions or user activities are recorded and used for intelligent content processing, e.g. employing collaborative filtering techniques. Here, the semantics is often represented implicitly, e.g. by computing similarity of users for relevant information artifact recommendation based on activities of similar users.

For large, dynamic or not completely known information content a lightweight semantics is often the only possible alternative to heavyweight representations, that offer advanced complex reasoning but they cannot be acquired automatically. Lightweight semantics representations form only basic conceptual structures [1]. We propose its representation based on a homogeneous underlying metadata

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 327–336, 2012.

representation: *knowledge tags* – metadata that describe an aspect of an information artifact, which can be either content-based or artifact's manipulation-based. Knowledge tag is an extension to the basic concept of a tag as a simple keyword or a term assigned to an information artifact. They represent any metadata that add additional value to the information artifact and represent some knowledge on web information system content. Knowledge tags for web-based information systems can contain explicit and implicit feedback generated by developers working on the source code [2].

Existing web information systems already assign metadata that can be considered as knowledge tags to documents – either manually or automatically [3]. By sharing of these knowledge tags, a new layer of the lightweight semantics over the web information system content can be created. In addition, the knowledge tags layer can become web information systems integration and sharing space based on metadata reuse either on the content itself or characteristics of its manipulation including user interaction or users. Web information systems can take metadata gained by other systems, use it for reasoning a new metadata and share it in the knowledge tags layer again. As a result, web information systems can collaboratively build and improve the semantic layer over the Web content.

However, existing systems obviously store the metadata in their private repositories, so other systems could not use the metadata. Moreover, an issue of dynamic change of information artifacts that can lead to invalidation of knowledge tags should be considered. We present an approach to knowledge tags maintenance, which allows systems to share their metadata in a consistent form via addressing issues related to:

- *The repository:* knowledge tags repository has to store a large amount of knowledge tags in a flexible open format which has to be understandable for web information systems and the repository has to still provide fast parallel access for a numbers of web information systems.
- *Dynamicity of the Web:* a content of the Web repositories is not stable. Stored documents arise, are deleted and modified without a notice. In addition, web users use the web content differently over the time. The Web content instability, diversity in usage of the Web content and also a time aspect can lead to invalidation of knowledge tags (e.g., new and favorite marks) that have to be updated or deleted.

We propose the knowledge tags repository and a method for storing and querying knowledge tags in it. For the repository design it is important to understand requirements for the repository, in particular automatic knowledge tags maintenance. We present also our presumptions on how third-party systems can use knowledge tags.

2 Related Work

Generally, there are two basic issues caused by dynamicity of the Web. Changes in tagged document can have influence to a content of knowledge tags. The influence of modifications in tagged documents to the content of knowledge tags is closely related to a type of knowledge tags and an algorithm, which created metadata stored in knowledge tags. Due to the complexity of change types identification and application to knowledge tags, knowledge tags are often deleted and recreated, although rebuild operations are time expensive and documents modifications require no or only small corrections of knowledge tags in the most of cases.

The second issue is knowledge tags' anchoring in documents, especially in textual documents that are frequently modified documents on the Web. In this case, the knowledge from annotation methods can be utilized, because of knowledge tags and annotations have common characteristics. Both of them are anchored to specific parts of documents and they contain small information on these document parts.

Popular methods of annotations anchoring are based on the start and the end letter indexes of an annotation. But this simple anchoring is not well-usable in dynamic documents on the Web, because web documents are changed without a notice and the index-based anchoring is not automatically repairable without change-set in documents. Moreover, the index-based anchoring is not able to reflex complex modifications, when a document was edited at the annotation's place and the modification has straight influence to both the anchoring and also annotation's content. In this case it is necessary to make decision if the anchoring would be updated or the annotation would become orphaned (new position of the annotation could not be assigned) and also how the annotation's content has to be updated [4].

Phelps and Wilensky proposed a method of robust annotations anchoring [5] with aim to start up development of an anchoring standard. They define criteria of anchoring robustness that are mainly focused on anchoring simplicity and its automatic correction based on new version of anchored document without necessity of a change-set. They also proposed their own anchoring based on SGDOM (simple, general document object model) which describes tree structure of textual documents. Every SGDOM logical part has its own identifier. Phelps and Wilensky utilize these identifiers and define three descriptors of a anchoring – *SGDOM identifier*, *path in a SGDOM tree* and *context*. Each descriptor is tolerant to different document change complexity on the expense of computational complexity.

iAnnotate tool [6] anchors users' annotations in webpages to DOM objects. iAnnotate does not reflect a webpage modifications, but it is focused on changes in the webpage presentation (e.g., a zoom or a change of resolution), to which iAnnotate easily reacts by obtaining new positions of DOM objects. iAnnotate stores annotations in a relational MySQL store which has good performance, but it is not easily distributable and it does not provide necessary flexibility for general annotations.

Anchoring representation based on a tree structure is used in Annotea system [7], too. HTML tree structure is utilized and anchoring descriptors defined by xPath. Annotea repository stores data in RDF model, which gives great flexibility to structure of annotations. Authors did not fully take this great advantage, but this approach to a repository inspired The Open Annotation Collaboration (www.openannotation.org) to a proposition of flexible open annotation model (www.openannotation.org/spec/beta), which allows storing and sharing annotations in unified form.

Annotations in this model consist of:

- *oac:Body*[1] – represents annotation's content which is stored in an annotation;
- *oac:Target* – represents target document which is annotated. An annotation can contain multiple targets;
- *oac:Constraint* – constrains annotation's body or target to their parts. A constraint on a body (an object of the type *oac:ConstrainedBody* derived from *oac:Body*) is

[1] Namespace oac: OA vocabulary (`http://www.openannotation.org/ns/`).

applicable, if only a part of a body is the real body of the annotation. Target constraint (an object of the type *oac:ConstrainedTarget* derived from *oac:Target*) specifies concrete part of the target, to which an annotation is anchored.

Some authors assign annotations only to concrete version of a document and they mark annotations as voided in each other version of a document. In work [8], authors proposed annotations maintenance method based on OA model and Memento framework [9], from which they determine versions of documents, for which annotations were created and after that they filter out voided annotations.

3 Knowledge Tags Maintenance

Current annotation systems support only specific types of annotation for specific types of target documents (e.g., text highlighting in HTML documents). They also provide basic annotations maintenance, mostly in a form of an annotations repository and an automatic anchoring repair based on predefined rules. We are working on knowledge tags maintenance approach to an automatized repair of knowledge tags after updating of tagged documents. A repair of a knowledge tag means discarding of a knowledge tag or updating its anchor and content. If a knowledge tag is not repairable, we mark the knowledge tag as voided and we yield decision how to modify the knowledge tag or if it have to be completely discarded to another system (if it is possible, the system which created the knowledge tag).

We address this goal via the method which consists of three main parts (Fig. 1):

- *Knowledge Tags Repository* – stores knowledge tags in flexible Open Annotation model;
- *Maintenance* – provides automatic maintenance over knowledge tags stored in the knowledge tags repository;
- *Access Provider* – provides access to the repository and notifies Maintenance about updates in the repository and detected new versions of tagged documents.

The maintenance part is responsible for the maintenance of knowledge tags consistency. It means that the maintenance guarantees for a correctness of knowledge tags (their anchoring, content and validity mark) that are provided to web information systems as a reaction to their requests (e.g., loading of knowledge tags anchored to a document). We achieve this via rules for knowledge tags maintenance. These rules are specific to each type of knowledge tags and they are defined by authors of concrete knowledge tag type. We also suppose that the rules are automatically derivable by watching of a knowledge tag life cycle and a life cycle of tagged documents.

The knowledge tags repository is core element, from which usability and performance of whole method is dependent. To achieve overall usability, the knowledge tags repository has to implement flexible and generally acceptable knowledge tags model and provide effective and powerful data access even with non-trivial amount of knowledge tags stored in it.

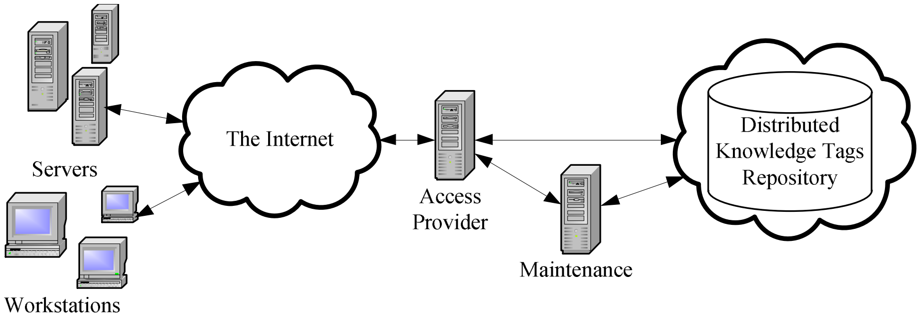


Fig. 1. Architecture of proposed knowledge tags maintenance approach

4 Knowledge Tags Repository

To supply acceptable knowledge tags model, we made decision to utilize existing Open Annotations model, which is currently in beta version but it is already used by a numbers of systems and projects [8], [10], [11]. The model is based on RDF and it is highly recommended to implement it by RDF triple databases and support at least a basic data access (e.g., querying by SPARQL) with RDF serialization output format.

To provide effective and powerful data access, we analyzed standard use cases of annotation repositories and itemize a list of requirements that respect these use cases and specific requirements of OA model and maintenance part of proposed method:

- *Storing of a knowledge tag* – creation of new knowledge tag for a document;
- *Updating of a knowledge tag* – e.g. after modification of a tagged document;
- *Obtaining of concrete knowledge tag* – retrieve the knowledge tag by its URI;
- *Access to knowledge tag's history* – obtaining of previous versions;
- *Obtaining of knowledge tags anchored to a document;*
- *Knowledge tags querying by SPARQL* – compatibility with OA model;
- *Distributed processing* – maintenance over non-trivial amount of knowledge tags.

Manipulation with the whole knowledge tag and not only with its parts is the main component of almost all standard use cases. It is a consequence of the fact that a knowledge tag has sense only as complete information with its content and also with its anchoring in tagged document. But this is in a disagreement with RDF triple databases that have good deduction possibilities but, they have serious issue with obtaining complete information about an object, when several simple queries have to be processed and each query can take several seconds in large datasets [12]. To address this issue, we set up hypothesis, that we can build efficient RDF-like repository for objects of one type (including types derived from this type) based on another than RDF triple stores, which allows efficient access to complete objects and also supports basic SPARQL query processing with a performance comparable to classical graph-based RDF stores.

4.1 Knowledge Tags Repository Structure

Document databases are in a correlation with our need of an access to whole knowledge tags, while they store documents (in general objects) as one item and not sparse over several tables or collections. This allows fast access to whole objects without necessity of time expensive joins [13]. We decided for MongoDB[2] which matches our requirements: it provides efficient data access (loading and updating) and supports distributed data processing via MapReduce [14].

MongoDB organizes stored objects in collections that allow developers to organize similar or related data to one collection. We design a structure of the knowledge tags repository based on two collections:

- *Tags* – contains knowledge tags in open structure which only have to meet with OA model. The collection provides fast access to knowledge tags by their URI, but access by URI of tagged document is inefficient, because the structure of OA model does not enable effective index over documents URIes;
- *Documents* – contains a mapping of documents to knowledge tags. The mapping has fixed format – a document URI, a knowledge tag URI, a validity of a knowledge tag and access rights. The fixed format allows fast filtrations and access to URIes of knowledge tags anchored to a document.

4.2 Distributed SPARQL Query Processing

MongoDB meets with almost all requirements to the repository. But it does not provide support for SPARQL query processing which is implementable via MapReduce. Several approaches to SPARQL query processing via MapReduce [15], [16] exists already, but all of them are proposed for Apache Hadoop[3] which has some differences in processing of Map and Reduce phases and proposed approaches work with RDF triples stores. MongoDB additionally provides Finalize function for efficient finalization of results of Reduce function.

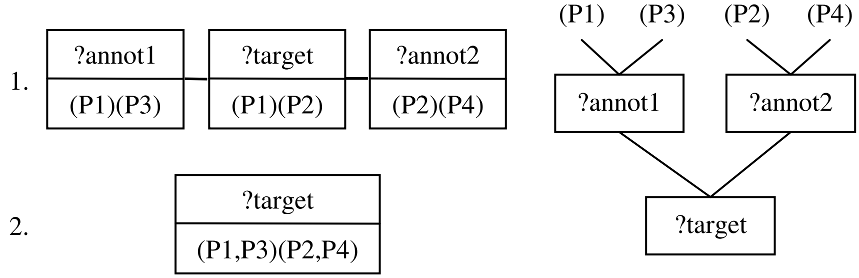


Fig. 2. Iterations of assembly algorithm (join graphs – left, join tree – right) for the example with four triple patterns: P1 – *?annot1 oac:hasTarget ?target*; P2 – *?annot2 oac:hasTarget ?target*; P3 – *?annot1 dcterms:creator ?creator1*; P4 – *?annot2 dcterms:creator ?creator2*

[2] http://www.mongodb.org/

[3] http://hadoop.apache.org/

Our algorithm for distributed SPARQL query processing firstly determines optimal join strategy to minimalize count of necessary join iterations via optimal join tree, the tree with minimal depth. Leafs of the optimal join tree are triple patterns and internal vertexes are join attributes. The tree assembly algorithm runs in a cycle until all join attributes are used in the tree. Each of iterations creates one layer of the tree (Fig. 2):

1. Create a join graph, the graph whose nodes are join attributes with a set of joined triple patterns (if two join attributes have equal set of covered triple patterns, these join attributes are represented by one node) and edges are inserted between nodes with intersecting set of joined triple patterns.
2. Until the join graph is not empty, select one node with the smallest degree and remove the node with incident nodes from the join graph.
3. Add join attributes from selected nodes to the optimal join tree and connect them to vertexes that are in previous layer and join common triple patterns.
4. Join triple pattern from selected nodes to new patterns.

The MapReduce algorithm uses an ordered list of join attributes with their values as a key and a list of deducted objects that consists of an ordered list of joined pattern ids and an ordered list of attributes from patterns with their values as a result. The algorithm is processed in two phases. The first phase is executed with join attributes on the lowest layer of the optimal join tree. In this phase Map function emits results from knowledge tags stored in the repository. Reduce function creates results as Cartesian products of Map results with same keys, where each of newly deducted objects contains complete list of pattern ids, from which it was built and a list of attributes and values from these patterns. Finalize function removes deducted objects that do not have complete list of patterns mapped to processed join keys (see Table 1).

Table 1. Examples of results of Map, Reduce and Finalize functions from the first phase

Function	Results

Map	{key : (annot1[X]), value : ({ (P1), (annot1[X] | target[page.html]) }) }
	{key : (annot1[X]), value : ({ (P3), (annot1[X] | creator1[John]) }) }
	{key : (annot1[Y]), value : ({ (P1), (annot1[Y] | target[style.css]) }) }

Reduce	{key : (annot1[X]), value : ({ (P1|P3), (annot1[X] | creator1[John] | target[page.html]) }) }
	{key : (annot1[Y]), value : ({ (P1), (annot1[Y] | target[style.css]) }) }

Finalize	{key : (annot1[X]), value : ({ (P1|P3), (annot1[X] | creator1[John] | target[page.html]) }) }

The second phase iteratively process remaining layers of the optimal join tree. Map function of this phase emits for each result from previous iteration new result with a key from current join attributes and unchanged value. Reduce and Finalize functions are same as in the first phase.

The SPARQL query processing algorithm is optimized for minimal count of join MapReduce cycles (one for each level of the optimal join tree), what decreases a number of necessary time expensive I/O operations between cycles.

5 Evaluation

To evaluate proposed repository, we realized knowledge tags repository solution based on MongoDB and repository based on Bigdata[4] RDF triple database powered by NanoSparqlServer. We selected Bigdata because of its performance and horizontal scalability, what makes possible to store non-trivial amount of knowledge tags. We also looked at in-memory databases (e.g. Trinity or JPregel) that have good performance but they have some issues with horizontal scalability and data persistence. For preliminary evaluation we deploy these repositories only on one node with Intel Core i7-640M processor, 8 GB DDR3 RAM and Seagate Momentus 7200.4 HDD.

During the evaluation we incrementally load one hundred of simple knowledge tags anchored to one (not same) document. Each of knowledge tags consists of sixteen RDF triples in OA model. After each load, we measured times of a load, a retrieving one knowledge tag by its URI, a retrieving URI list of knowledge tags anchored to a document, a retrieving knowledge tags anchored to a document and an execution of simple SPARQL query with one join attribute.

Measured values oscillate around linear function. These oscillations were mainly caused by background system processes and make impossible straight comparison of measured times. For this reason we made linear transposition function of measured values (e.g., Fig. 3) and compared transposed values.

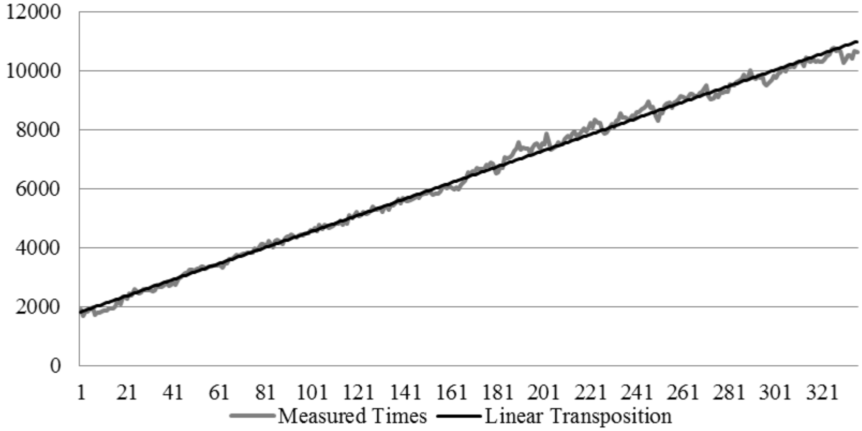


Fig. 3. The dependency of incremental data load duration (in milliseconds) to Bigdata database from a number of knowledge tags (in hundreds) in the repository

The comparison of these two repository realizations shows that the proposed solution based on MongoDB is more effective than the repository based on Bigdata. It is mostly visible on primary operations over knowledge tags. These operations were from 400 to 600 times faster (Fig. 4). Very important is also that less important operation, SPARQL query evaluation, took approximately same time in both realizations.

[4] `http://www.systap.com/bigdata.htm`

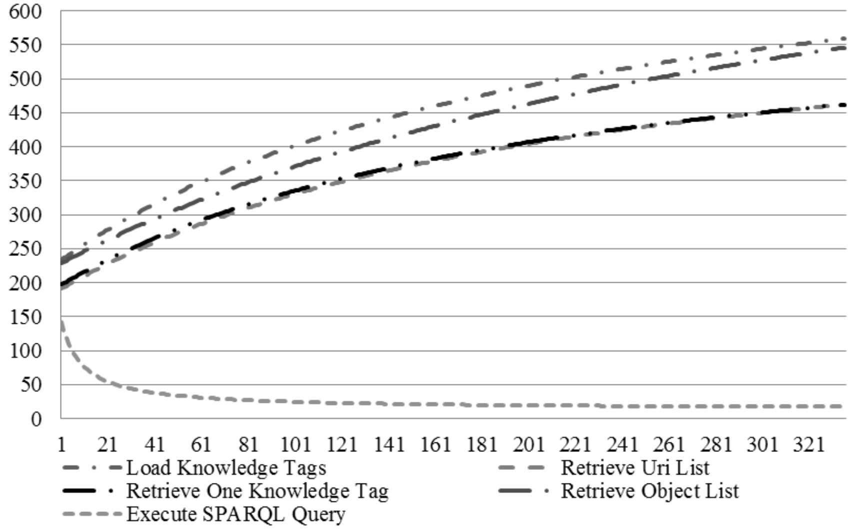


Fig. 4. The comparison of Bigdata and MongoDB realizations. The Y axis presents measured times ratio (how many times is the Bigdata realization slower than the MongoDB realization) and the x axis presents hundreds of knowledge tags stored in the repository in a measuring case.

6 Conclusions and Future Work

In this paper we have presented novel concept of employing knowledge tags for representation of semantics over the web information content. We concentrate on maintaining knowledge tags within proposed knowledge tags repository based on Open Annotation model and realized by MongoDB. The knowledge tags represent lightweight semantics, which includes not only already used content annotations such as relevant domain terms or concepts in educational content [3] or named objects on pictures [17], but also other indices representing various characteristics related to particular information artifact, e.g. its popularity, aggregated users' activity logs or inferred characteristics such as quality. The knowledge tags in that manner create a collaboration space of the web information system, where several web information systems can reuse existing knowledge tags to infer new knowledge tags and so to enrich and improve lightweight semantics over the information space [2].

We also present results of preliminary performance evaluations of the repository. These results indicate that proposed repository is much effective than classical RDF triple stores for our use cases. But for general confirmation of our hypothesis we have to evaluate our approach with more RDF triple databases (including in-memory databases) distributed to several nodes.

Our next steps lead to a proposal of a rule engine based on MapReduce for automatic knowledge tags maintenance. The rule engine employs machine learning to automatically deduce new rules for the maintenance of knowledge tags and improve existing rules by watching of knowledge tags life cycle. These rules are not independent (modification of one knowledge tag can lead to a necessity of modification of several other knowledge tags), so the rule engine should provide effective strategy of rules evaluation.

Acknowledgements. This work was partially supported by the grants VG1/0675/11, VG1/0971/11 and it is the partial result of the Research & Development Operational Programme for project ITMS 26240220039, co-funded by the ERDF.

References

1. Bieliková, M., Barla, M., Šimko, M.: Lightweight Semantics for the "Wild Web". In: IADIS Int. Conf. WWW/Internet 2011. IADIS Press (2011)
2. Bieliková, M., et al.: Collaborative Programming: The Way to "Better" Software. In: 6th Workshop on Int. and Knowledge Oriented Tech., Košice, pp. 89–94 (2011) (in Slovak)
3. Šimko, M.: Automated Acquisition of Domain Model for Adaptive Collaborative Web-Based Learning. Inf. Sciences and Tech., Bulletin of the ACM Slovakia 2(4), 9 p. (2012)
4. Priest, R., Plimmer, B.: RCA: Experiences with an IDE Annotation Tool. In: 6th ACM SIGCHI New Zealand Chapter's Int. Conf. on Computer-human Interaction Design Centered HCI, pp. 53–60. ACM Press, New York (2006)
5. Phelps, T.A., Wilensky, R.: Robust Intra-Document Locations. Computer Networks 33, 105–118 (2000)
6. Plimmer, B., et al.: iAnnotate: Exploring Multi-User Ink Annotation in Web Browsers. In: 9th Australasian Conf. on User Interface, pp. 52–60. Australian Comp. Soc. (2010)
7. Kahan, J., Koivunen, M.R.: Annotea: An Open RDF Infrastructure for Shared Web Annotations. In: 10th Int. Conf. on WWW, pp. 623–632. ACM Press, New York (2001)
8. Sanderson, R., Van de Sompel, H.: Making Web Annotations Persistent over Time. In: 10th Annual Joint Conf. on Digit. Lib., pp. 1–10. ACM Press, New York (2010)
9. Van de Sompel, H., Nelson, M.L., Sanderson, R., Balakireva, L.L., Ainsworth, S., Shankar, H.: Memento: Time Travel for the Web. CoRR. abs/0911.1, p. 14 (2009)
10. Gerber, A., Hyland, A., Hunter, J.: A Collaborative Scholarly Annotation System for Dynamic Web Documents – A Literary Case Study. In: Chowdhury, G., Koo, C., Hunter, J. (eds.) ICADL 2010. LNCS, vol. 6102, pp. 29–39. Springer, Heidelberg (2010)
11. Yu, C.H., Groza, T., Hunter, J.: High Speed Capture, Retrieval and Rendering of Segment-Based Annotations on 3D Museum Objects. In: Xing, C., Crestani, F., Rauber, A. (eds.) ICADL 2011. LNCS, vol. 7008, pp. 5–15. Springer, Heidelberg (2011)
12. Rohloff, K., Dean, M., Emmons, I., Ryder, D., Sumner, J.: An Evaluation of Triple-Store Technologies for Large Data Stores. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2007, Part II. LNCS, vol. 4806, pp. 1105–1114. Springer, Heidelberg (2007)
13. Tiwari, S.: Professional NoSQL. John Wiley & Sons, Inc., Indianapolis (2011)
14. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM 51, 107–113 (2008)
15. Kim, H.S., Ravindra, P., Anyanwu, K.: From SPARQL to MapReduce: The Journey Using a Nested TripleGroup Algebra. VLDB Endowment 4, 1426–1429 (2011)
16. Myung, J., Yeon, J., Lee, S.: SPARQL Basic Graph Pattern Processing with Iterative MapReduce. In: Workshop on Massive Data Analytics on the Cloud, p. 6. ACM Press, New York (2010)
17. Kuric, E.: Automatic Photo Annotation Based on Visual Content Analysis. Information Sciences and Technologies. Bulletin of the ACM Slovakia 2(3), 72–75 (2011)

Investigating the Relation between Users' Cognitive Style and Web Navigation Behavior with K-means Clustering

Marios Belk, Efi Papatheocharous, Panagiotis Germanakos, and George Samaras

Department of Computer Science, University of Cyprus, Nicosia, Cyprus
{belk,efi.papatheocharous,pgerman,cssamara}@cs.ucy.ac.cy

Abstract. This paper focuses on modeling users' cognitive style based on a set of Web usage mining techniques on navigation patterns and clickstream data. Main aim is to investigate whether k-means clustering can group users of particular cognitive style using measures obtained from a series of psychometric tests and content navigation behavior. Three navigation metrics are proposed and used to find identifiable groups of users that have similar navigation patterns in relation to their cognitive style. The proposed work has been evaluated with a user study which entailed a psychometric-based method for extracting the users' cognitive styles, combined with a real usage scenario of users navigating in a controlled Web environment. A total of 22 participants of age between 20 and 25 participated in the reported study providing interesting insights with respect to cognitive styles and navigation behavior of users.

1 Introduction

The World Wide Web today has expanded to serve millions of different users for a multitude of purposes in all parts of the world. Naturally, Web content nowadays needs to be filtered and personalized based on the particular needs of individual users. The users' interests, expectations and expertise, cognitive style and perception are some of the factors that need to be considered when creating personalized interactive systems. Therefore, the first step towards Web personalization is specifying the type of users and creating the user model which reflects the user's intrinsic needs and preferences which ultimately influence the adaptation of Web interactive systems.

The user model is a representation of static and dynamic information about an individual, and it represents an essential entity for an adaptive interactive system aiming to provide adaptation effects (i.e., a set of tasks or content of a system can be presented differently between users with different user models) [1]. For example, an adaptive information retrieval system may recommend the top most relevant items based on the user's interests. An adaptive educational hypermedia system may provide adjusted educational material and navigation support to users that have particular level of knowledge on a subject. An adaptive e-commerce system may enhance the security and privacy-preserving measures and can present an adapted content to users that have a specific level of knowledge and experience towards security terms (e.g., provide novice users with personalized security awareness information by using

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 337–346, 2012.

simplified security terms and additional explanations). The mechanism utilized for user modeling can be based on explicit or implicit information gathering approaches. Explicit information is provided directly by the user, usually through Web registration forms, questionnaires, or specially designed psychometric instruments. On the other hand, implicit information is extracted by the system automatically to infer characteristics about the user and is usually obtained by tracking the user's navigation behavior throughout the system. For example, such implicit information can be extracted from the time spent on a particular Web-page by a user, which can be used to infer the interest of the user towards the main subject of that Web-page.

To this end, the work presented studies the relation between users' cognitive styles and navigation behavior with explicit and implicit user information gathering approaches. Main objectives of the paper are to: i) investigate whether a specific clustering technique (i.e., *k*-means clustering) can group users of particular cognitive style using measures obtained from psychometric tests, ii) propose navigation content metrics to find identifiable groups of users that have similar navigation patterns in relation to their cognitive style, and iii) investigate whether there is a possible relationship between users' cognitive style and their navigation behavior. The identification of users with specific cognitive and navigation style will ultimately help in defining an adaptation mechanism that will target a different user interface experience in Web-based environments for various cognitive typologies of users.

2 User Modeling Based on Data Analysis Techniques

The ability of adaptation in interactive systems heavily depends on successful user modeling. A user model is created through a user modeling mechanism in which unobservable information about a user is inferred from observable information from that user [2]; for example, using the interactions with the system (i.e., time being active on a Web-page, buying history, ratings of products, bookmarked or saved content, etc.).

The simplest approach of user model generation is in the case where the information collected by the user is used as-is and remains unprocessed. For example, users might explicitly express their interest on specific topics of a news publishing system which will be further used by simple rule-based mechanisms to adapt the interface by displaying the selected topics on the top of the user's interface. More intelligent approaches for generating user models is in the case where the browsing activities of users may be utilized by data mining and machine learning techniques to recognize regularities in user paths and integrate them in a user model. A thorough literature review on how data mining techniques can be applied to user modeling in the context of personalization systems can be found in [3]. The data mining techniques mentioned enable pattern discovery through clustering and classification, association rules (or association discovery) and sequence mining (or sequential pattern discovery). They represent popular approaches appearing in the data mining literature. In addition, [4] describes data mining algorithms based on clustering, association rule discovery, sequential pattern mining, Markov models and probabilistic mixture and hidden (latent) variable models for Web personalization purposes.

Nowadays, the process of Web user modeling has become attached to automated data mining or knowledge discovery techniques due to the large volumes of available user data on the Web [5]. Nasraoui et al. [5] perform clustering on user sessions to place users in homogeneous groups based on the similar activities performed and then extract specific user profiles from each cluster. Clustering techniques are also used in order to divide users into segments containing users with similar navigation behavior. Using a similarity metric, a clustering algorithm groups the most similar users together to form clusters. Some algorithms classify users into multiple segments and describe the strength of each relationship [6]. The same concept is found within fuzzy clustering techniques, examples of which include the work of Castellano and Torsello [7] that categorized users based on the evaluation of similarity between fuzzy sets using a relational fuzzy clustering algorithm and Castellano et al. [8] that derived user profiles by analyzing user interests. Variations of fuzzy clustering methods include Fuzzy c-medoids, Fuzzy c-trimmed-medoids, relational Fuzzy Clustering-Maximal Density Estimator (RFC-MDE) algorithm and hierarchical clustering approaches.

The abovementioned works primarily focus on applying data mining and machine learning techniques for modeling the interests and preferences of users towards specific items of Web environments. For example, clustering techniques are utilized for grouping users that visited, bought or rated similarly the same products. Association rules are used in many cases to relate different products based on their viewing history, e.g., when users view product A and afterwards view product B, then an association rule is created between product A and B indicating a high relationship between the two products. Accordingly, this information is further utilized by the system to offer recommendations based on the navigation behavior of users.

Taking into consideration these works, the next section presents a user modeling approach for eliciting similar groups of users based on their navigation behavior in the context of adaptive interactive systems and relates these groups to cognitive styles. To the best of the authors' knowledge, this is among the first works to study the relation between the cognitive style of users and their navigation behavior in an online encyclopedia system, apart from sporadic attempts which utilized a number of clustering techniques to understand human behavior and perception in relation with cognitive style, expertise and gender differences of digital library users [9], and recent research attempts which studied the connection between the way people move in a museum and the way they prefer to approach and process information cognitively [10].

3 Cognitive-Based User Modeling for Web Adaptation

The proposed approach, as shown in Figure 1, focuses on the user modeling part of an adaptive interactive system. User modeling is associated to information regarding the users of a system, the users' interactions as well as the context in which communication or data transaction takes place. It is mainly responsible for gathering information regarding the user, building the user model and feeding this information to the adaptation mechanism which in turn will modify the user interface accordingly.

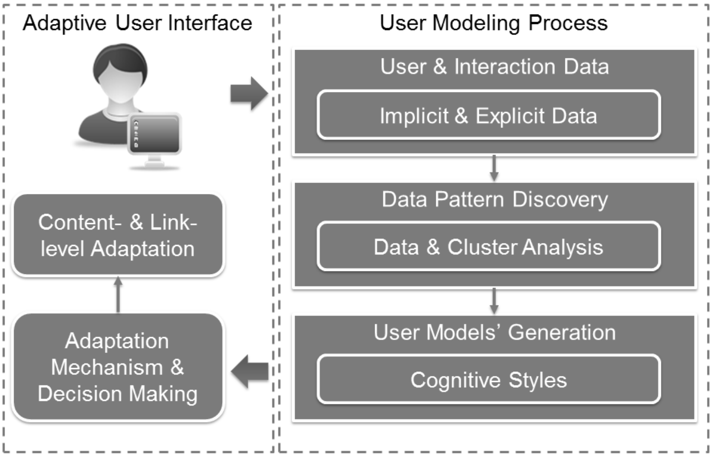


Fig. 1. Cognitive-based User Modeling Approach

Based on Figure 1, the first step of the proposed approach starts with collecting the user's interaction data with the system. Specifically, the browsing history is used as the main source of information about the user's interaction data which contains the URLs visited by the user and the date/time of the visits. Accordingly, meaningful information is derived based on this information, e.g., the number of visits to a particular URL, the time spent and the specific sequence of visits. In the next step, specific data analysis and clustering techniques are applied on the raw data in order to classify users to groups with similar navigation behavior and extract other important information about the users. Finally, the user models are generated containing information about the cognitive style of users which is further provided to the adaptation mechanism to apply the adaptation effects. The next section makes a brief introduction of the theory behind the cognitive styles utilized in this work.

3.1 Cognitive Styles

Cognitive styles represent an individually preferred and habitual approach to organizing and representing information [11]. Among numerous proposed theories of individual styles [11-13], this study utilizes Riding's Cognitive Style Analysis (CSA) [11]. In particular, Riding's CSA consists of two dimensions and classifies users to the cognitive typologies of Wholist-Intermediate-Analyst and Verbal-Intermediate-Imager. The Wholist-Analyst dimension refers to how individuals organize information. Specifically, users that belong to the Wholist class view a situation and organize information as a whole and are supposed to take a linear approach in hypermedia navigation (i.e., users read the material in a specific order based on the context). On the other hand, users that belong to the Analyst class view a situation as a collection of parts, stress one or two aspects at a time and are supposed to take a non-linear approach in hypermedia navigation. Users that belong in between the two end points (i.e., Intermediate) do not differ significantly with regards to information organization. The Verbal-Imager dimension refers to how individuals process information. Users that belong to the Verbal class can proportionally process textual and/or auditory content more efficiently than images, whereas users that belong to the Imager class the opposite. Users that belong in between the two end points (i.e., Intermediate) do not differ significantly with regards to information processing.

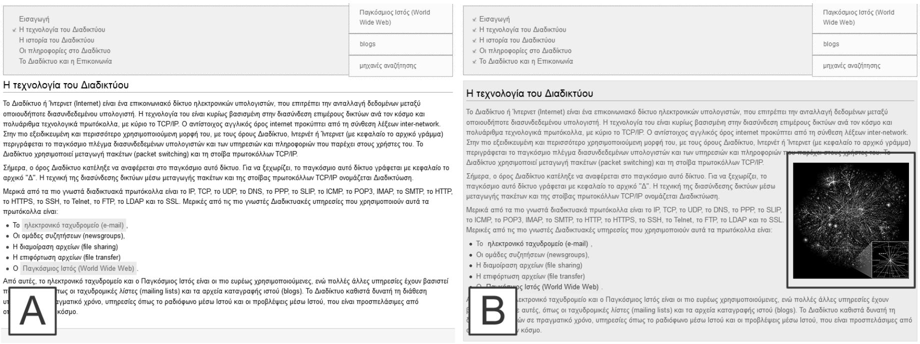
Riding's CSA might be applied effectively on designing adaptive hypermedia systems, since it consists of distinct scales that correspond directly to different aspects of information systems, i.e., content and functionality is either presented visually or verbally, and users may have specific navigation behavior, i.e. linear vs. non-linear, based on their cognitive style.

4 User Study

The objective of the study is threefold; firstly, investigate whether clustering techniques can group users of particular cognitive style using measures obtained from Riding's CSA test, secondly, evaluate the use of content navigation metrics to find identifiable groups of users that have similar navigation patterns within the group of users that participated in the study, and finally investigate whether a relation exists between cognitive style and navigation behavior of users.

4.1 Method of Study

A total of 22 individuals participated voluntarily in the study carried out within the first week of November 2011. All participants were undergraduate Computer Science students in their third and fourth year of study, and their age varied from 20 to 25. The students first completed a series of questions using a Web-based psychometric test (http://adaptiveweb.cs.ucy.ac.cy/profileConstruction) based on Riding's CSA [11] that measures the response time on two types of stimuli and computes a ratio between the response times for each stimuli type in order to highlight differences in cognitive style. The stimuli types are: a) statements (i.e., identify whether a statement is true or false), and b) pictures (i.e., compare whether two pictures are identical, and whether one picture is included in the other). Then, the users were asked to read various articles of a Web environment and navigate freely through its hyperlinks. Main aim was to track the navigation sequence of users within the Web environment. Accordingly, an appropriate environment for tracking the navigation sequence of users is a reproduced version of Wikipedia (http://wikipedia.org) since it consists of content and hyperlinks that are placed in a context-dependent order and thus enables users either navigate sequentially, or in an unordered form. Furthermore, the articles were enriched to include verbal-based content, i.e., content in textual form without images (Figure 2A), or image-based content, i.e., content represented with images and diagrams (Figure 2B). The same content was presented to all users, while verbal-based and imager-based content was presented to users that belonged to the Verbal class and the Imager class, respectively.


Εισαγωγή
Η τεχνολογία του Διαδικτύου
Η ιστορία του Διαδικτύου
Οι πληροφορίες στο Διαδίκτυο
Το Διαδίκτυο και η Επικοινωνία
Παγκόσμιος Ιστός (World Wide Web)
blogs
μηχανές αναζήτησης
A
B

Η τεχνολογία του Διαδικτύου

Το Διαδίκτυο ή Ίντερνετ (Internet) είναι ένα επικοινωνιακό δίκτυο ηλεκτρονικών υπολογιστών, που επιτρέπει την ανταλλαγή δεδομένων μεταξύ οποιουδήποτε διασυνδεδεμένου υπολογιστή. Η τεχνολογία του είναι κυρίως βασισμένη στην διασύνδεση επιμέρους δικτύων ανά τον κόσμο και πολυάριθμα τεχνολογικά πρωτόκολλα, με κύριο το TCP/IP. Ο αντίστοιχος αγγλικός όρος internet προκύπτει από τη σύνθεση λέξεων inter-network. Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, με τους όρους Διαδίκτυο, Ιντερνέτ ή Ίντερνετ (με κεφαλαίο το αρχικό γράμμα) περιγράφεται το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και των υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Το Διαδίκτυο χρησιμοποιεί μεταγωγή πακέτων (packet switching) και τη στοίβα πρωτοκόλλων TCP/IP.

Σήμερα, ο όρος Διαδίκτυο κατέληξε να αναφέρεται στο παγκόσμιο αυτό δίκτυο. Για να ξεχωρίζει, το παγκόσμιο αυτό δίκτυο γράφεται με κεφαλαίο το αρχικό "Δ". Η τεχνική της διασύνδεσης δικτύων μέσω μεταγωγής πακέτων και της στοίβας πρωτοκόλλων TCP/IP ονομάζεται Διαδικτύωση.

Μερικά από τα πιο γνωστά διαδικτυακά πρωτόκολλα είναι το IP, TCP, το UDP, το DNS, το PPP, το SLIP, το ICMP, το POP3, IMAP, το SMTP, το HTTP, το HTTPS, το SSH, το Telnet, το FTP, το LDAP και το SSL. Μερικές από τις πιο γνωστές Διαδικτυακές υπηρεσίες που χρησιμοποιούν αυτά τα πρωτόκολλα είναι:

- Το ηλεκτρονικό ταχυδρομείο (e-mail) ,
- Οι ομάδες συζητήσεων (newsgroups),
- Η διαμοίραση αρχείων (file sharing)
- Η επιφόρτωση αρχείων (file transfer)
- Ο Παγκόσμιος Ιστός (World Wide Web) .

Από αυτές, το ηλεκτρονικό ταχυδρομείο και ο Παγκόσμιος Ιστός είναι οι πιο ευρέως χρησιμοποιούμενες, ενώ πολλές άλλες υπηρεσίες έχουν βασιστεί π τως οι ταχυδρομικές λίστες (mailing lists) και τα αρχεία καταγραφής ιστού (blogs). Το Διαδίκτυο καθιστά δυνατή τη διάθεση υ αγματικό χρόνο, υπηρεσίες όπως το ραδιόφωνο μέσω Ιστού και οι προβλέψεις μέσω Ιστού, που είναι προσπελάσιμες από ο κόσμο.

Η τεχνολογία του Διαδικτύου

Το Διαδίκτυο ή Ίντερνετ (Internet) είναι ένα επικοινωνιακό δίκτυο ηλεκτρονικών υπολογιστών, που επιτρέπει την ανταλλαγή δεδομένων μεταξύ οποιουδήποτε διασυνδεδεμένου υπολογιστή. Η τεχνολογία του είναι κυρίως βασισμένη στην διασύνδεση επιμέρους δικτύων ανά τον κόσμο και πολυάριθμα τεχνολογικά πρωτόκολλα, με κύριο το TCP/IP. Ο αντίστοιχος αγγλικός όρος internet προκύπτει από τη σύνθεση λέξεων inter-network. Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, με τους όρους Διαδίκτυο, Ιντερνέτ ή Ίντερνετ (με κεφαλαίο το αρχικό γράμμα) περιγράφεται το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και των υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Το Διαδίκτυο χρησιμοποιεί μεταγωγή πακέτων (packet switching) και τη στοίβα πρωτοκόλλων TCP/IP.

Σήμερα, ο όρος Διαδίκτυο κατέληξε να αναφέρεται στο παγκόσμιο αυτό δίκτυο. Για να ξεχωρίζει, το παγκόσμιο αυτό δίκτυο γράφεται με κεφαλαίο το αρχικό "Δ". Η τεχνική της διασύνδεσης δικτύων μέσω μεταγωγής πακέτων και της στοίβας πρωτοκόλλων TCP/IP ονομάζεται Διαδικτύωση.

Μερικά από τα πιο γνωστά διαδικτυακά πρωτόκολλα είναι το IP, TCP, το UDP, το DNS, το PPP, το SLIP, το ICMP, το POP3, IMAP, το SMTP, το HTTP, το HTTPS, το SSH, το Telnet, το FTP, το LDAP και το SSL. Μερικές από τις πιο γνωστές Διαδικτυακές υπηρεσίες που χρησιμοποιούν αυτά τα πρωτόκολλα είναι:

- Το ηλεκτρονικό ταχυδρομείο (e-mail) ,
- Οι ομάδες συζητήσεων (newsgroups),
- Η διαμοίραση αρχείων (file sharing)
- Η επιφόρτωση αρχείων (file transfer)
- Ο Παγκόσμιος Ιστός (World Wide Web) .

Α λεκτρονικό ταχυδρομείο και ο Παγκόσμιος Ιστός είναι οι πιο ευρέως χρησιμοποιούμενες, ενώ πολλές άλλες υπηρεσίες έχουν β ε αυτές, όπως οι ταχυδρομικές λίστες (mailing lists) και τα αρχεία καταγραφής ιστού (blogs). Το Διαδίκτυο καθιστά δυνατή τη δ ών σε πραγματικό χρόνο, υπηρεσίες όπως το ραδιόφωνο μέσω Ιστού και οι προβλέψεις μέσω Ιστού, που είναι προσπελάσιμες από σ ν κόσμο.

Fig. 2. Verbal-based (A) and Image-based (B) User Interface of the Web-site used in the Study

The navigation behavior of the students was monitored at all times on the client-side. In particular, a browser-based logging facility was implemented with JQuery JavaScript Library (http://jquery.com) to collect the client-side usage data from the hosts accessing the Web-site used for the study.

4.2 Definition of Metrics

The reproduced version of Wikipedia consists of articles that are interconnected through hyperlinks based on a context-dependent hierarchy (i.e., articles of similar context are interconnected). We consider that sequential links are connected based on the articles' content and the distance between each point is equal to 1. Thus, a linear navigation behavior is represented with a minimum distance covered considering the links visited by a user whereas a higher distance describes a non-linear navigation behavior. Accordingly, we measure the distance between the links visited by users utilizing the following metrics: i) Absolute Distance of Links (ADL), the total absolute distance between the links visited, ii) Average Sequential Links (ASL), the average number of sequential links visited by a user, and iii) Average non-Sequential Groups of Links (AGL), the average number of non-sequential groups of links visited by a user, if all sequential links are considered to represent one group. To better explain the metrics used, we provide a navigation example, e.g., the clickstream navigation pattern "Node: 8, Pat: 4 | Node: 9, Pat: 3-2 |" which means that the user visited the eighth content-page and then read the content of the fourth link and so on. For this particular navigation the metrics, as defined above, are calculated as: ADL=(|4-1|+|3-1|+|2-3|)/N=2, ASL=M/N=0.333 and AGL=B/N=0.667, where N is the number of total links visited, M is the number of sequential links visited based on the Web-site content and B is the number of non-sequential groups of links derived from the links visited. In our example, M is equal to 1 and B is equal to 2 as for pattern "3-2" the only sequential link clicked was the second and the two non-sequential groups of links were patterns "4" and "3-2". The user's interaction with the Web-site content was captured through these metrics which were also normalized based on the number of user interactions by dividing each variable to the total number of clicks.

4.3 Results

This section presents and analyzes the results obtained from the study. A non-hierarchical method based on the Euclidean distance (*k*-means clustering) was used [14]. The following methods were applied: i) *k*-means clustering on the responses of users to the psychometric test, and ii) *k*-means clustering on the navigation pattern of users in the online encyclopedia system.

Since the data obtained was from different users, and thus generated independently, we may assume that the i.i.d. assumption holds. Moreover, since the possible navigation patterns and user interactions with the user interface were close to a very large number, *k*-means clustering was selected for the analysis to avoid calculating all possible distances between all possible interactions. Other assumptions made was that the structure of the Web-site's is linear based on the content (i.e., it contains an introduction and sections that follow the introduction in a sequential manner) and that the number of clusters is known in each case (i.e., *k*=3 and *k*=2 in each clustering case respectively). Thus, using *k*-means clustering we are trying to differentiate users based on their Cognitive Style (CS) typology (i.e., Wholist-Intermediate-Analyst and Verbal-Intermediate-Imager) and navigation style (i.e., linear and non-linear).

Table 1. Ratio of Cognitive-based Profiles of Clustered Users from the Psychometric Test

Cluster	Users	Wholist-Analyst Range	Users	Verbal-Imager Range
1	8	[0.786, 1.030]	6	[1.121, 1.248]
2	12	[1.099, 1.424]	7	[0.958, 1.040]
3	2	[1.776, 1.853]	9	[0.832, 0.941]

Table 1 presents the number of clustered users in each cluster using *k*-means and the range of ratios obtained from the psychometric tests. From the clustering performed using the users' responses in the psychometric test, we observe that the users are clustered in three groups. The figures show that the users are clearly distinguished based on their cognitive profile and that they cover the whole range of the scale as suggested by Riding [11]. For example, in the Wholist-Analyst dimension one of the clusters contains users with CS ratio [0.786, 1.030] which is in line to Riding's Wholist typology (i.e., ≤1.02) and in the Verbal-Imager dimension the clustered users' CS ratio [0.832, 0.941] is again in line to Riding's Verbal typology (i.e., ≤0.98). This finding shows that the *k*-means clustering technique can group users of particular CS using measures obtained from psychometric tests; it has provided encouraging results and justifies further utilization.

The next clustering applied involves content visit path analysis by using the three metrics proposed which measure the linearity of the users' navigation behavior. Table 2 presents the ranges of the users' cognitive-based profiles appearing in each cluster. For example, the CS of the users grouped based on their content navigation style in the first cluster is within the range [0.819, 1.776] regarding the Wholist-Analyst dimension and within the range [0.861, 1.248] regarding the Verbal-Imager dimension.

Table 2. Ratio of Cognitive-based Profiles of Clustered Users from Content Navigation Style

Metric	Ranges of Cluster 1		Ranges of Cluster 2	
	Wholist-Analyst	Verbal-Imager	Wholist-Analyst	Verbal-Imager
ADL	[0.819, 1.776]	[0.861, 1.248]	[0.786, 1.853]	[0.832, 1.154]
ASL	[0.819, 1.248]	[0.861, 1.121]	[0.786, 1.853]	[0.832, 1.248]
AGL	[0.819, 1.776]	[0.885, 1.248]	[0.786, 1.853]	[0.832, 1.128]
All	[0.819, 1.776]	[0.861, 1.248]	[0.786, 1.853]	[0.832, 1.128]

The normalized values of the metrics were used to perform clustering with the combination of all three clustering metrics (results of which are shown in the last row of Table 2) and to also visualize the degree of membership of each user per metric in each cluster (Figure 3).

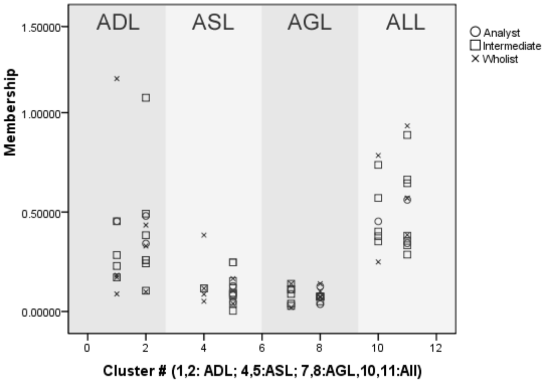


Fig. 3. Degree of Membership in each Cluster (in columns) per Metric ADL, ASL and AGL and CS Identification regarding the Wholist-Analyst Dimension

Table 2. Mann-Whitney Rank-sum Statistical Test per Clustering Metric

Metric	Cluster 1			Cluster 2		
	U	z	p	U	z	p
ADL	63	0.6	0.274	47	0.46	0.322
ASL	66	-1.6	0.054	54	-0.66	0.254
AGL	43	0.62	0.267	28	1.7	0.044

From Figure 3 we observe that the users grouped in each cluster present variability in terms of their CS. In addition, the results of the rank-sum statistical test (Mann-Whitney [15]) performed between the two clusters (Table 3), has shown that in most cases the two clusters did not differ significantly. However, we observed that using the ASL and AGL metrics, statistically significant differences were identified between the first and the second clusters' medians for the Wholist-Analyst and Verbal-Imager ratios respectively (U=66, p=0.05, and U=28, p=0.04). This means that the ASL and AGL metric proposed can be used to identify users of particular navigation behaviour that differ in their Wholist-Analyst and Verbal-Imager ratio styles respectively. Conclusively, users (in Cluster 1) with linear navigation behavior have statistically significant different cognitive style ratio concentration than non-linear users (in Cluster 2), which is an important result, since it can be further concluded that some relation has been found between navigation and cognitive styles in these particular cases.

4.4 Final Remarks

Based on the results obtained, currently, no safe conclusion can be drawn, whether there is a cohesive correlation between the cognitive style and the navigation pattern followed by each user, and further experimentation needs to be carried out. In particular, most users in the same cluster although had similar navigation behavior (i.e., linear/non-linear), their respective cognitive style was variant. However, a statistical comparison of the CS ratios of users between the clusters showed that the users' cognitive style differed significantly ($p \leq 0.05$) indicating that clustering users based on their navigation behavior is likely to cause separation of users into distinctive groups that differ significantly in terms of cognitive style. In addition, the navigation metrics proposed seem to successfully distinguish clusters of users that according to their respective cognitive-based profile range belong to two overlapping groups – range that covers a lower and a higher fragment in the Riding CSA scale. Finally, the resulting membership degree to each cluster can be used to characterize the degree of linearity in the interaction of users in fuzzy terms to optimally capture navigation behavior. Such findings could provide a promising future direction towards modeling cognitive styles of users by tracking their navigation behavior with implicit information gathering approaches that are transparent to the users, as well as the identification of adaptation rules for a more user-centric interface design.

5 Conclusions and Future Work

This paper investigated the relation between cognitive style and navigation behavior of users. Specific navigation metrics have been proposed and utilized by a clustering technique, with the aim to identify groups of users that have similar navigation behavior and investigate the relation to their cognitive style.

The limitations of the current work are related to the small sample of users participating in the study, the number of clusters used in the analysis (for example in larger samples perhaps a higher number of clusters should be used), the selection of clustering method, the effect of outliers and the order of cases analyzed. We have addressed some of these threats by assuming to know how many clusters we wanted to extract based on the cognitive profiles of the users and since we also had a moderately sized dataset this meant that the selection of *k*-means clustering was a reasonable choice. The solution of *k*-means clustering may depend on the order of the users using the online environment and thus we have arranged the samples in a random order to address this threat. In addition, the fact that the *k*-means clustering algorithm is sensitive to the initial randomly selected cluster centers, we have eliminated this threat by repeating the algorithm execution several times.

The relevant research is in its infancy and further empirical studies are needed to investigate such issues. A future research prospect is to evaluate other Web environments, techniques (e.g., fuzzy clustering and neural networks) and metrics that might also assess human behavior in order to shed more light on this complex phenomenon.

Acknowledgements. This work is co-funded by the EU project CONET (INFSO-ICT-224053), SocialRobot (285870) and the project smarTag (University of Cyprus).

References

1. Brusilovsky, P., Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007)
2. Frias-Martinez, E., Magoulas, G., Chen, S., Macredie, R.: Modeling Human Behavior in User-Adaptive Systems: Recent Advances Using Soft Computing Technique. J. Expert Systems with Applications 29(2), 320–329 (2005)
3. Eirinaki, M., Vazirgiannis, M.: Web Mining for Web Personalization. J. ACM Transactions on Internet Technology 3(1), 1–27 (2003)
4. Mobasher, B.: Data Mining for Web Personalization. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 90–135. Springer, Heidelberg (2007)
5. Nasraoui, O., Soliman, M., Saka, E., Badia, A., Germain, R.: A Web Usage Mining Framework for Mining Evolving User Profiles in Dynamic Web Sites. J. IEEE Transactions on Knowledge and Data Engineering 20(2), 202–215 (2008)
6. Perkowitz, M., Etzioni, O.: Adaptive Web Sites. Communications of the ACM 43(8), 152–158 (2000)
7. Castellano, G., Torsello, M.A.: Categorization of Web Users by Fuzzy Clustering. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 222–229. Springer, Heidelberg (2008)
8. Castellano, G., Fanelli, A., Mencar, C., Torsello, M.: Similarity-Based Fuzzy Clustering for User Profiling. In: Conferences on Web Intelligence and Intelligent Agent Technology Workshops, pp. 75–78. IEEE Computer Society, Washington, USA (2007)
9. Frias-Martinez, E., Chen, S., Macredie, R., Liu, X.: The Role of Human Factors in Stereotyping Behavior and Perception of Digital Library Users: A Robust Clustering Approach. J. User Modeling and User-Adapted Interaction 17(3), 305–337 (2007)
10. Antoniou, A., Lepouras, G.: Modeling Visitors' Profiles: A Study to Investigate Adaptation Aspects for Museum Learning Technologies. J. Computing Cultural Heritage 3(2), 1–19 (2010)
11. Riding, R., Cheema, I.: Cognitive styles - An Overview and Integration. J. Educational Psychology 11(3/4), 193–215 (1991)
12. Felder, R., Silverman, L.: Learning and Teaching Styles in Engineering Education. J. Engineering Education 78(7), 674–681 (1988)
13. Witkin, H., Moore, C., Goodenough, D., Cox, P.: Field-dependent and Field-independent Cognitive Styles and their Educational Implications. Review of Educational Research 47(1), 1–64 (1977)
14. Aldenderfer, M., Blashfield, R.: Cluster Analysis. Sage Publications, Newbury Park (1984)
15. Mann, H., Whitney, D.: On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other. J. Annals of Mathematical Statistics 1(8), 50–60 (1947)

Web Navigation via Semantic Annotations

Roberto De Virgilio and Lorenzo Dolfi

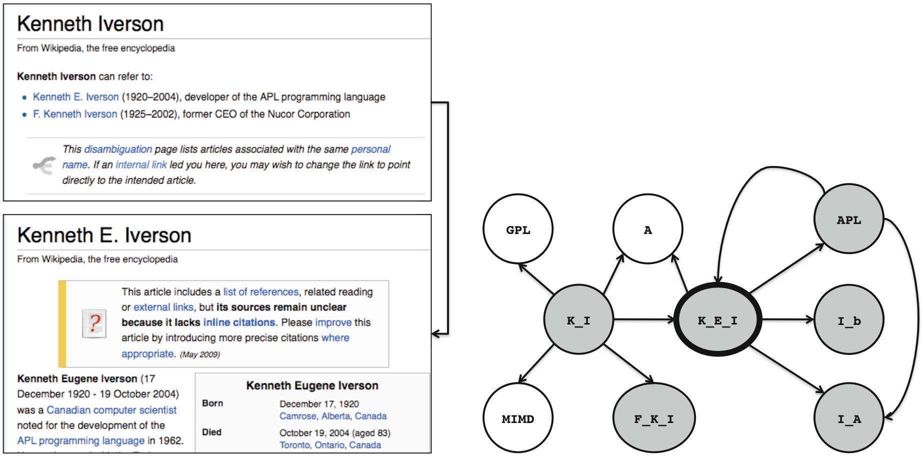
Dipartimento di Informatica e Automazione,
Universitá Roma Tre, Rome, Italy
{dvr,dolfi}@dia.uniroma3.it

Abstract. Searching relevant information from Web may be a very tedious task. If people cannot navigate through the Web site, they will quickly leave. Thus, designing effective navigation strategies on Web sites is crucial. In this paper we provide and implement centrality indices to guide the user for an effective navigation of Web pages. We get inspiration from well-know location family problems to compute the center of a graph: a joint use of such indices guarantees the automatic selection of the best starting point. To validate our approach, we have developed a system that implements the techniques described in this paper on top of an engine for keyword-based search over RDF data. Such system exploits an interactive front-end to support the user in the visualization of both annotations and corresponding Web pages. Experiments over widely used benchmarks have shown very good results.

1 Introduction

The original perception of the Web by the vast majority of its early users was as a static repository of unstructured data. This was reasonable for browsing small sets of information by humans, but this static model now breaks down as programs attempt to dynamically generate information, and as human browsing is increasingly assisted by intelligent agent programs. With the size and availability of data constantly increasing, a fundamental problem lies in the difficulty users face to find and retrieve the information they are interested in [9]. A relevant problem is that using a search engine probably the retrieved pages are not always what user is looking for. To give an example, let us consider Wikipedia and type "`Kenneth Iverson APL`", i.e., we would retrieve information about the developer of APL programming language. As shown Fig. 1.(a), the user has to browse the list of all Kenneth Iverson, then manually to solve the disambiguation (i.e., selecting Kenneth E. Iverson) and finally to consume the information about development of APL programming language following the link in the Web page. Such task is time consuming and in case of a long chain of links to follow it could convince the user to quickly leave the Web site. To this aim Semantic Web helps encoding properties of and relationships between objects represented by information stored on the Web [4]. It envisions that authors of pages include semantic information along with human-readable Web content, perhaps with some machine assistance in the encoding. Referring to the example of Fig. 1.(a), we could annotate the corresponding Web pages with the following RDF triples

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 347–357, 2012.



Kenneth Iverson

From Wikipedia, the free encyclopedia

Kenneth Iverson can refer to:

- Kenneth E. Iverson (1920–2004), developer of the APL programming language
- F. Kenneth Iverson (1925–2002), former CEO of the Nucor Corporation

This disambiguation page lists articles associated with the same personal name. If an internal link led you here, you may wish to change the link to point directly to the intended article.

Kenneth E. Iverson

From Wikipedia, the free encyclopedia

This article includes a list of references, related reading or external links, **but its sources remain unclear because it lacks inline citations**. Please improve this article by introducing more precise citations where appropriate. *(May 2009)*

Kenneth Eugene Iverson (17 December 1920 - 19 October 2004) was a Canadian computer scientist noted for the development of the APL programming language in 1962.

Kenneth Eugene Iverson	
Born	December 17, 1920 Camrose, Alberta, Canada
Died	October 19, 2004 (aged 83) Toronto, Ontario, Canada

(a) (b)

Fig. 1. (a) Searching K.E. Iverson, developer of the APL programming language, and (b) Semantic Annotation of K. Iverson wikipedia pages

```
<rdf:Description rdf:about="wiki:Kenneth_Iverson">
 <rdf:type rdf:resource="wiki:Article"/>
 <wiki:redirectsTo rdf:resource="wiki:Kenneth_E._Iverson"/>
 <wiki:redirectsTo rdf:resource="wiki:F._Kenneth_Iverson"/>
        ...
</rdf:Description>

<rdf:Description rdf:about="wiki:Kenneth_E._Iverson">
 <rdf:type rdf:resource="wiki:Article"/>
 <wiki:internalLink rdf:resource="wiki:APL_programming_language"/>
 <wiki:internalLink rdf:resource="wiki:Iverson_Award"/>
 <wiki:internalLink rdf:resource="wiki:Iverson_bracket"/>
        ...
</rdf:Description>
```

Due to the huge amount of RDF data available on the Web (currently around 7 billion RDF triples and 150 million RDF links), keywords search based systems (e.g., [5]) are increasingly capturing the attention of researchers, implementing IR strategies on top of traditional database management systems with the goal of freeing the users from having to know data organization and query languages. The aim of such approaches is to query semantic annotations (in terms of RDF graphs [1]) instead of making a deep analysis of a large number of Web pages. The result is a set of RDF subgraphs linked to the Web pages of interest. For instance Fig. 1.(b) shows the RDF subgraph matching the query "Kenneth Iverson APL". For the sake of simplicity, we used only initials of URIs and marked matching nodes in gray. However most of the proposals do not analyze in depth how to exploit the resulting RDF annotation to navigate the pages of interest. The user manually has to analyze the RDF graph, selecting the starting node from which begins the navigation of the corresponding Web pages. Of course, in this situation semi-automatic tools would support the analysis but the risk is to guide the user to select a wrong starting point, so far from the most interesting Web pages. For instance looking at the RDF graph of Fig. 1.(b), a semi-automatic tool could select K_I as starting point (i.e., it is the source of the graph): in this case we reconduct the user to the same situation of Fig. 1.(a). The best choice would be K_E_I linking to the Web page of Kenneth E. Iverson.

In this paper we provide and implement centrality indices to guide the user for an effective navigation of Web pages. We get inspiration from well-know location family problems to compute the center of a graph: a joint use of such indices guarantees the automatic selection of the best starting point. To validate our approach, we have developed a system that implements the techniques described in this paper on top of an engine for keyword-based search over RDF data. Such system exploits an interactive front-end to support the user in the visualization of both annotations and corresponding Web pages. Experiments over widely used benchmarks have shown very good results, in terms of both effectiveness and efficiency. The rest of the paper is organized as follows. In Section 2 we discuss the related research. In Section 3 we present our centrality indices. Finally Section 4 illustrates the experimental results, and in Section 5, we draw our conclusions and sketch future works.

2 Related Work

Facility location analysis deals with the problem of finding optimal locations for one or more facilities in a given environment. Location problems are classical optimization problems with many applications in industry and economy. The spatial location of facilities often take place in the context of a given transportation, communication, or transmission system, which may be represented as a network for analytic purposes. A first paradigm for location based on the minimization of transportation costs was introduced by Weber [15] in 1909. However, a significant progress was not made before 1960 when facility location emerged as a research field. There exist several ways to classify location problems. According to Hakimi [7] who considered two families of location problems we categorize them with respect to their objective function. The first family consists of those problems that use a minimax criterion. As an example, consider the problem of determining the location for an emergency facility such as a hospital. The main objective of such an emergency facility location problem is to find a site that minimizes the maximum response time between the facility and the site of a possible emergency. The second family of location problems considered by Hakimi optimizes a minisum criterion which is used in determining the location for a service facility like a shopping mall. The aim here is to minimize the total travel time. A third family of location problems described for example in [13] deals with the location of commercial facilities which operate in a competitive environment. The goal of a competitive location problem is to estimate the market share captured by each competing facility in order to optimize its location. Our focus here is not to treat all facility location problems. The interested reader is referred to a bibliography devoted to facility location analysis [6]. The aim of this paper is to introduce three important vertex centralities by examining location problems.

Then we can introduce a fourth index based not only on “spatial” properties (such as the other centrality indices) but also on the semantics. The definition of different objectives leads to different centrality measures. A common feature, however, is that each objective function depends on the distance between the vertices of a graph. In the following we focus on G as connected directed graph with at least two vertices and we suppose that the distance $d(u, v)$ between two vertices u and v is defined as the length of the shortest path from u to v. These assumptions ensure that the following centrality indices are well defined. Moreover, for reasons of simplicity we consider G to be an unweighted graph, i.e., all edge weights are equal to one. Of course, all indices presented here can equally well be applied to weighted graphs.

3 Web Navigation

As said in the Introduction, user is supported by different approximate query processing methods to improve the search of information on the Web. In particular Semantic Web was introduced to annotate the semantics involved into a Web page, making more automatic the interoperability between applications and machines and improving the effectiveness of the results. However a significant issue is to exploit the result (annotation) of the query processing to navigate the corresponding Web pages in an effective way. Formally, the result can be modeled as a labelled directed graph G. It is a six element tuple $G = \{V, E, \Sigma_V, \Sigma_E, L_G, \omega\}$ where V is a set of vertices and $E \subseteq V \times V$ is a set of ordered pairs of vertices, called edges. Σ_V and Σ_E are the sets of vertices and edge labels, respectively. The labelling function L_G defines the mappings $V \rightarrow \Sigma_V$ and E $\rightarrow \Sigma_E$, while the weight function ω assigns a (positive) score to each node by defining the mapping $V \rightarrow \mathbb{N}$. Then, centrality indices can be computed to quantify an intuitive feeling that in the result some vertices or edges are more central than others. Such indices can support the user to navigate directly the part of the result that best fits the query provided by the user.

3.1 Center Indices

In the following we get inspiration from well-know location family problems to compute the center of G.

Eccentricity. The aim of the first problem family is to determine a location that minimizes the maximum distance to any other location in the network. Suppose that a hospital is located at a vertex $u \in V$. We denote the maximum distance from u to a random vertex v in the network, representing a possible incident, as the eccentricity $e(u)$ of u, where $e(u) = max\{d(u, v) : v \in V\}$. The problem of finding an optimal location can be solved by determining the minimum over all $e(u)$ with $u \in V$. In graph theory, the set of vertices with minimal eccentricity is denoted as the center of G. Hage and Harary [8] proposed a centrality measure based on the eccentricity $c_E(u) = \frac{1}{e(u)} = \frac{1}{max\{d(u,v):v \in V\}}$ This measure is consistent with the general notion

Algorithm 1. Center computation by eccentricity

```
    Input  : The graph $G$
    Output: The center $c_E$

1   $n \leftarrow V.length$;
2   $L_E \leftarrow$ InitializeArray($n$);
3   $M \leftarrow$ FloydWarshall($G$);
4   for $i \leftarrow 0$ to $n$ do
5       $L_E[i] \leftarrow$ Max($M[i]$);
6   $i_{min} \leftarrow$ MinIndex($L_E$);
7   $c_E \leftarrow V[i_{min}]$;
8   return $c_E$;
```

of vertex centrality, since $e(u)^{-1}$ grows if the maximal distance of u decreases. Thus, for all vertices $u \in V$ of the center of G: $c(u) \geqslant c_E(v)$ for all $v \in V$. Based on such method, we define a procedure to compute the center of the graph as described in the Algorithm 1. In the algorithm, by using the function `InitializeArray`, we initialize the eccentricity vector L_E (line[2]). Such vector has length n (the number of nodes in V): for each node with index i we calculate the maximum distance from the nodes of G (lines[4-5]). The distances from each couple of nodes are computed in a matrix M (line[3]) by using the Floyd-Warshall algorithm [11], that is a graph analysis algorithm for finding shortest paths in a weighted graph. If there does not exist a path between two nodes we set the distance to ∞. Finally we select the index i_{min} in L_E corresponding to the minimum value (line[6]). The center c_E corresponds to the node with the index i_{min} in V. For instance let us consider the graph of Fig. 1.(b). The matrix computed by Floyd-Warshall algorithm is

$$M = \begin{bmatrix} 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty & \infty & \infty & \infty \\ 1 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 2 \\ \infty & \infty & \infty & \infty & 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & 1 & \infty & \infty & 0 & 1 & 1 & 1 \\ \infty & \infty & 2 & \infty & \infty & 1 & 0 & 2 & 1 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

where idx(`GPL`) = 1, idx(`MIMD`) = 2, idx(`A`) = 3, idx(`K_I`) = 4, idx(`F_K_I`) = 5, idx(`K_E_I`) = 6, idx(`APL`) = 7, idx(`I_b`) = 8 and idx(`I_A`) = 9. Then the eccentricity vector L_E is $L_E = \begin{bmatrix} \infty & \infty & \infty & 2 & \infty & \infty & \infty & \infty & \infty \end{bmatrix}^t$. In L_E the minimum value is 2, corresponding to the index 4: in this case the center c_E is `K_I`.

Closeness. Next we consider the second type of location problems - the minisum location problem, often also called the median problem or service facility location problem. Suppose we want to place a service facility, e.g., a shopping mall, such that the total distance to all customers in the region is minimal. This would make traveling to the mall as convenient as possible for most customers. We denote the sum of the distances from a vertex $u \in V$ to any other vertex in a graph G as the total distance $\sum_{v \in V} d(u, v)$. The problem of finding an appropriate location can

Algorithm 2. Center computation by closeness

```
  Input  : The graph $G$
  Output: The center $c_C$

1 $n \leftarrow V.length$;
2 $L_C \leftarrow$ InitializeArray($n$);
3 $M \leftarrow$ FloydWarshall($G$);
4 for $i \leftarrow 0$ to $n$ do
5     $L_C[i] \leftarrow \sum_{j=0}^{n-1} M[i][j]$;
6 $i_{min} \leftarrow$ MinIndex($L_C$);
7 $c_C \leftarrow V[i_{min}]$;
8 return $c_C$;
```

be solved by computing the set of vertices with minimum total distance. In social network analysis a centrality index based on this concept is called closeness. The focus lies here, for example, on measuring the closeness of a person to all other people in the network. People with a small total distance are considered as more important as those with a high total distance. Various closeness-based measures have been developed, see for example [2,3,10,12] and [14]. The most commonly employed definition of closeness is the reciprocal of the total distance $c_C(u) = \frac{1}{\sum_{v \in V} d(u,v)}$. In our sense this definition is a vertex centrality, since $c_C(u)$ grows with decreasing total distance of u and it is clearly a structural index. Before we discuss the competitive location problem, we want to mention the radiality measure and integration measure proposed by Valente and Foreman [14]. These measures can also be viewed as closeness-based indices. They were developed for digraphs but an undirected version is applicable to undirected connected graphs, too. This variant is defined as $c_R(u) = \frac{\sum_{v \in V}(\triangle_G + 1 - d(u,v))}{n-1}$, where $\triangle_G$ and n denote the diameter of the graph and the number of vertices, respectively. The index measures how well a vertex is integrated in a network. The better a vertex is integrated the closer the vertex must be to other vertices. The primary difference between c_C and c_R is that c_R reverses the distances to get a closeness-based measure and then averages these values for each vertex. Based on such method, we define a procedure to compute the center of the graph as described in the Algorithm 2. As for the eccentricity, we initialize the closeness vector L_C and calculate the matrix M. Then for each node with index i we calculate the sum of distances from the other nodes (lines[4-5]). Finally, as for the eccentricity, we calculate the index i_{min} of the minimum value in L_C. Such index corresponds to the center c_C in G. Referring again to our example, given the matrix M by the Floyd-Warshall algorithm, we have the following closeness vector $L_C = \left[\infty\ \infty\ \infty\ 11\ \infty\ \infty\ \infty\ \infty\ \infty\right]^t$. Since the minimum value is 11, i_{min} is 4: also in this case the center is K_I.

Centroid Values. The last centrality index presented here is used in competitive settings. Suppose each vertex represents a customer in a graph. The service location problem considered above assumes a single store in a region. In reality, however, this is usually not the case. There is often at least one competitor

Algorithm 3. Center computation by centroid values

```
Input  : The graph $G$
Output: The center $c_F$

$n \leftarrow V.length$;
$C \leftarrow$ InitializeMatrix($n$,$n$);
$min \leftarrow$ InitializeArray($n$);
$M \leftarrow$ FloydWarshall($G$);
for $i \leftarrow 0$ to $n$ do
    for $j \leftarrow 0$ to $n$ do
        if $i == j$ then
            $C[i][j] \leftarrow \infty$;
        else
            for $h \leftarrow 0$ to $n$ do
                if $h \neq i \wedge h \neq j$ then
                    if $M[i][h] < M[j][h]$ then
                        C[i][j] $\leftarrow$ C[i][j] +1;
                    else if $M[i][h] > M[j][h]$ then
                        C[i][j] $\leftarrow$ C[i][j] -1;
for $i \leftarrow 0$ to $n$ do
    min[i] $\leftarrow$ Min($C[i]$);
$i_{max} \leftarrow$ MaxIndex($min$);
$c_F \leftarrow V[i_{max}]$;
return $c_F$;
```

offering the same products or services. Competitive location problems deal with the planning of commercial facilities which operate in such a competitive environment. For reasons of simplicity, we assume that the competing facilities are equally attractive and that customers prefer the facility closest to them. Consider now the following situation: a salesman selects a location for his store knowing that a competitor can observe the selection process and decide afterwards which location to select for her shop. Which vertex should the salesman choose? Given a connected undirected graph G of n vertices. For a pair of vertices u and v, $\gamma_u(v)$ denotes the number of vertices which are closer to u than to v, that is $\gamma_u(v) = |\{w \in V : d(u,w) < d(v,w)\}|$. If the salesman selects a vertex u and his competitor selects a vertex v, then he will have $\gamma_u(v) + \frac{1}{2}(n - \gamma_u(v) - \gamma_v(u)) = \frac{1}{2}n + \frac{1}{2}(\gamma_u(v) - \gamma_v(u))$ customers. Thus, letting $f(u,v) = \gamma_u(v) - \gamma_v(u)$, the competitor will choose a vertex v which minimizes $f(u,v)$. The salesman knows this strategy and calculates for each vertex u the worst case, that is $c_F(u) = min\{f(u,v) : v \in V - u\}$. $c_F(u)$ is called the centroid value and measures the advantage of the location u compared to other locations, that is the minimal difference of the number of customers which the salesman gains or loses if he selects u and a competitor chooses an appropriate vertex v different from u. Based on such method, we define a procedure to compute the center of the graph as described in the Algorithm 3. In the algorithm, we initialize the centroid vector min and the centroid matrix C, i.e., $n \times n$, where each value [i,j] corresponds to $f(i,j)$. We fill C (lines[5-15]) by using the matrix M, calculated by the Floyd-Warshall algorithm. Then for each row i of C we copy the minimum value in $min[i]$ (lines[16-17]). Finally we calculate the index i_{max} corresponding to the maximum value in min (line[18]). The center c_F correspond to the node in V with index i_{max}. Referring again to our example

$$C = \begin{bmatrix} \infty & 0 & 0 & -7 & 0 & -4 & -4 & 0 & 0 \\ 0 & \infty & 0 & -7 & 0 & -4 & -4 & 0 & 0 \\ 0 & 0 & \infty & -7 & 0 & -4 & -4 & 0 & 0 \\ 7 & 7 & 7 & \infty & 7 & 0 & 3 & 7 & 7 \\ 0 & 0 & 0 & -7 & \infty & -4 & -4 & 0 & 0 \\ 4 & 4 & 4 & 0 & 4 & \infty & 2 & 4 & 4 \\ 4 & 4 & 4 & -3 & 4 & -2 & \infty & 4 & 4 \\ 0 & 0 & 0 & -7 & 0 & -4 & -4 & \infty & 0 \\ 0 & 0 & 0 & -7 & 0 & -4 & -4 & 0 & \infty \end{bmatrix}$$

from C we compute the following vector $min = \begin{bmatrix} -7 & -7 & -7 & 0 & -7 & 0 & -3 & -7 & -7 \end{bmatrix}^t$. In this case the maximum value is 0, corresponding to two indexes: 4 and 6. This means that we have two centroids, i.e., K_I and K_E_I.

3.2 Effective Navigation of Web Pages

The methods discussed above compute the center of a graph with respect to the topology information on the nodes. Referring to the example in Fig. 1.(b), in any method we have the center K_I (the centroid method reports K_E_I also). In this case such center allows to reach all nodes of the graph, but the navigation starting from such center is not effective: K_I corresponds to the Web page with all Kenneth Iverson. The best starting point would be K_E_I that is the Kenneth Iverson directly linked to the APL programming language page. Therefore beyond the center based on the spatial information of the graph, we need a "center of interest", i.e., some vertex that is more closed to the significant pages than others. In other words we need the node, with no-zero score, that is close to the nodes having high scores (i.e., matching the keywords). Therefore, we define a procedure to compute the center of interest as described in the Algorithm 4. In the algorithm, we calculate the closeness of each node, that is the sum of distances from the others but we normalize it with respect to the score of the node (lines[5-13]): the center of interest will have the minimum closeness with the highest score (i.e., ω refers to the function in [5]). If the node with index i we are considering has score 0 then the closeness is ∞. We store all values into the vector D, initialized by InitializeArray(line[2]). Finally we calculate the index i_{min} corresponding to the minimum value in D and the center of interest c will be the node in V with index i_{min}. Referring again to our example we have the following vector $D = \begin{bmatrix} \infty & \infty & \infty & \frac{8}{2} & \frac{12}{1} & \frac{6}{2} & \frac{9}{1} & \frac{10}{1} & \frac{9}{1} \end{bmatrix}^t$. Since the minimum value is 3 (i.e., $\frac{6}{2}$) the center of interest has index 6 (i.e., K_E_I). The joint use of the center calculated by spatial methods and the center of interest allows an effective navigation.

Algorithm 4. Center of interest

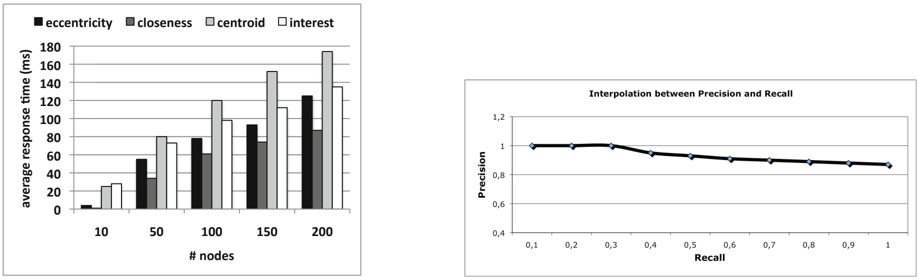
```
Input  : The graph $G$
Output: The center of interest $c$

$n \leftarrow V.length$;
$D \leftarrow$ InitializeArray($n$);
$M \leftarrow$ FloydWarshall($G$);
min $\leftarrow 0$;
for $i \leftarrow 0$ to $n$ do
    $D[i] \leftarrow 0$;
    if $\omega(V[i]) > 0$ then
        for $j \leftarrow 0$ to $n$ do
            if $\omega(V[j]) > 0$ then
                $D[i] \leftarrow D[i] + M[i][j]$;
        $D[i] \leftarrow \frac{D[i]}{\omega(V[i])}$;
    else
        $D[i] \leftarrow \infty$;
$i_{min} \leftarrow$ MinIndex($D$);
$c \leftarrow V[i_{min}]$;
return $c$;
```

4 Experimental Results

We implemented our framework in a Java tool[1]. The tool is according to a client-server architecture. At client-side, we have a Web interface based on GWT that provides (i) support for submitting a query, (ii) support for retrieving the results and (iii) a graphical view to navigate the Web pages via the resulting annotation. At server-side, we have the core of our query engine. We used YAANII [5], a system for keyword search over RDF graphs. We have executed several experiments to test the performance of our tool. Our benchmarking system is a dual core 2.66GHz Intel with 2 GB of main memory running on Linux. We have used Wikipedia3, a conversion of the English Wikipedia into RDF. This is a monthly updated data set containing around 47 million triples. The user can submit a keyword search query Q to YAANII that returns the top-10 solutions. Each solution is a connected subgraph of Wikipedia3 matching Q. In Fig. 2.(a) we show the performance of our system to compute the centrality indices. In particular we measured the average response time (ms) of ten runs to calculate the center in any method. Then, publishing the system on the Web, we asked to several and different users (i.e., about 100) to test the tool by providing a set of ten keyword search queries and to indicate if the centers are really effective. In this way we calculate the interpolation between precision and recall as shown in Fig. 2.(b). All these results validate the feasibility and effectiveness of our approach.

[1] A video tutorial is available at http://www.youtube.com/watch?v=CpJhVhx3r80



(a) (b)

Fig. 2. Performance (a) and Effectiveness (b) of the approach

5 Conclusion and Future Work

In this paper we discussed and implemented an approach for an effective navigation of Web pages by using semantic annotations. The approach is based on defining and implementing centrality indices, allowing the user to automatically select the starting point from which to reach the Web pages of interest. Experimental results demonstrate how significant it is the use of semantic annotations for surfing the Web effectively. In particular, an effective visualization of the annotation matching the user request improves the quality of the navigation. For future work, we are investigating new methods to determine the starting point and an implementation for distributed architectures (e.g., mobile environment).

References

1. Angles, R., Gutierrez, C.: Querying RDF Data from a Graph Database Perspective. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 346–360. Springer, Heidelberg (2005)
2. Bavelas, A.: Communication patterns in task oriented groups. Journal of the Acoustical Society of America 22(6), 271–282 (1950)
3. Beauchamp, M.A.: An improved index of centrality. Behavioral Science 10(2) (1965)
4. De Virgilio, R., Giunchiglia, F., Tanca, L. (eds.): Semantic Web Information Management - A Model-Based Perspective. Springer (2010)
5. De Virgilio, R., Cappellari, P., Miscione, M.: Cluster-Based Exploration for Effective Keyword Search over Semantic Datasets. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 205–218. Springer, Heidelberg (2009)
6. Domschke, W., Drexl, A.: Location and Layout Planning: An International Bibliography. Springer, Berlin (1985)
7. Hakimi, S.L.: Optimum location of switching centers and the absolute centers and medians of a graph. Operations Research 12(2), 450–459 (1964)
8. Harary, F., Hage, P.: Eccentricity and centrality in networks. Social Networks 17(1), 57–63 (1995)
9. Li, W.-S., Candan, K.S., Vu, Q., Agrawal, D.: Retrieving and organizing web pages by “information unit”. In: WWW, pp. 230–244. ACM Press (2001)

10. Moxley, R.L., Moxley, N.F.: Determining point-centrality in uncontrived social networks. Sociometry 37(1), 122–130 (1974)
11. Rosen, K.H.: Discrete Mathematics and Its Applications. Addison Wesley (2003)
12. Sabidussi, G.: The centrality index of a graph. Psychometrika 31(4), 581–603 (1966)
13. Smart, C., Slater, P.J.: Center, median and centroid subgraphs. Networks 34(4), 303–311 (1999)
14. Valente, T.W., Foreman, R.K.: Measuring the extent of an individual's connectedness and reachability in a network. Social Networks 20(1), 89–105 (1998)
15. Weber, A.: Uber den Standort der Industrien. J. C. B. Mohr, Tubingen (1909)

Using the Web to Monitor a Customized Unified Financial Portfolio

Camilo Restrepo-Arango, Arturo Henao-Chaparro, and Claudia Jiménez-Guarín

Department of Systems and Computing Engineering,
Universidad de los Andes,
111711, Bogotá, Colombia
{c.restrepo235,a.henao59,cjimenez}@uniandes.edu.co

Abstract. Unified Financial Portfolio is a financial application that applies concepts of information retrieval and Web 2.0 to provide a better understanding of the market trends taking into account the users information needs. It integrates and retrieves non-structured content related to the financial domain, from social networks, Stock Exchange and newspapers. The main contributions of this work are the software architecture, the content based semantic search model and the use of Big Data technology. The model is based on the vector-space model, including a retrieval weighting algorithm using domain specific considerations. This paper presents the application evaluation, using a large amount of unstructured content from dynamic and social web content sources.

Keywords: Web 2.0, Sentiment Analysis, Social Content, NoSQL database, Information Retrieval.

1 Introduction

Nowadays the amount of financial information is growing very fast, being published on different places in different formats, which makes it unstructured, diverse, heterogeneous, changing and hard to understand for users who are drawn to these subjects. This means it is difficult for the interested people to relate it and to exploit it to the maximum potential. Many Web sites like Google Finance [1], Yahoo! Finance [2], and Bloomberg [3], offers online financial services and information for general or informed people. They cover from basic training to specialized content delivery: users can, for example, create a portfolio, view the market quotes of their stocks and the latest news, sometimes related to it. Bloomberg also offers market trends, and some free services like Bloomberg TV or Bloomberg Radio. However, the financial market is wide and there are many types of investments which information can be scattered all over different sources that have to be found.

Even if several potential sources can be used to satisfy such wide range of needs, handling different kind of sources, such as journals, magazines, stock exchange portals and subscription services, among others, can be essential in order to offer interesting information, undiscovered relationships among sources and meaningful information integration. Despite having all these alternatives, it is the user's responsibility to gather

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 358–367, 2012.

and filter the relevant material, in a time-consuming and hard to achieve task. Web 2.0 [4] approaches provide tools for managing dynamic information and make easier to take advantage of collective knowledge generation. This includes not only financial facts, but also what people are publishing, deciding or commenting in social media.

The main contribution of this article is an information retrieval model focused on the financial sector, enabling integration and retrieval of relevant Web information for a specific user given his interests using a Web information retrieval approach [5] [6]. It is a mash-up consolidating 242 financial information sources and uses the recollected content as the main basis for retrieval. The paper is organized as follows. Section 2 reviews related work; Section 3 describes our proposal, the Unified Financial Portfolio (UFP). Section 4 presents the information retrieval model for UFP. Next, the implementation, test and validation features are presented. Finally, conclusions and future work are suggested.

2 Related Work

Web applications like Google Finance and Yahoo! Finance allow users to follow the information associated with their portfolios in a dynamic and pleasant way, providing dynamic information like events, stock prices variations and news that influence the market trends. They also relate the stock prices changes and the published news, allowing users to understand recent variations of prices. However, it is not possible to include the perception that social networks, like Twitter or Facebook, could provide about people's point of view regarding companies strategic decisions.

The stock recommendation engine Stoocker [7] uses stock market prices and financial news to present user portfolios composition, including prices and news for each company in which the user has investments, but it does not take into account news about other relevant related companies. Stoocker uses a recommendation system based on the community perception of the market based on the number of user submissions, which can lead to inaccurate results used to make important decisions.

Hermes [8] is an ontology based framework that is able to read news and extract economic information. It stores the most important concepts in the domain of interest and updates itself, allowing to detect changes in the real world. The presented implementation allows the user selects his concepts of interest which are used to perform queries. The results show a good performance in terms of precision and recall. However, it only uses news as input, it is not clear if it can handle multi-language news and concepts and it doesn't take into account social media content.

StockWatcher [9] is an OWL-based application to extract relevant news items from financial RSS feeds. It is user customizable and it requires four steps to access the desired news. The user must specify the companies, the information and the newsfeeds he want to monitor. The presented results don't provide any measure like precision or recall to allow a comparison and it has the same disadvantages as Hermes.

YourNews [10] is an application for news access personalization. It takes into account the user behavior to recommend content extracted from newsfeeds. This application deals mostly with ease of use and user personalization. It uses a classical vector space model to index news content. The presented results use measures like precision and recall; time spent reading articles and average ranks of user clicks. This application has the same disadvantages as the two previous presented works.

SemNews [11] is an application to monitor RSS newsfeeds and provide a structured representation of the meaning of news. Their strength is the use of a dynamic ontology to represent concepts. Features like sources retrieval using content, or sources other than RRS feeds are not specified in the paper.

Beside these well-known specialized sites, many online newspapers[1] offer to the wide public general financial information, usually not customizable and without considering social networks. In the academic context, several works present models [12], applications and mash-ups [13], but they rarely integrate multi-language information sources, dealing with social, non-structured data and financial context.

3 Unified Financial Portfolio

UFP is a specific domain application for financial information integration and retrieval considering user portfolio and related social content, to a wide user's spectrum. It offers relevant news related to portfolios, including the market reaction extracted from social media, and assisting the decision making process for each portfolio asset.

3.1 General Description

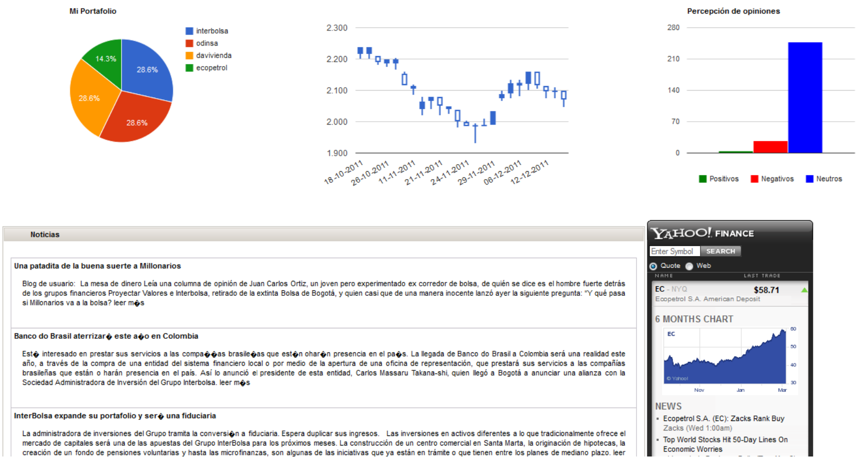
UFP presents information about user financial portfolios, including news, detailed financial information, and social trends related to each portfolio asset. From user defined financial information filters, UFP gathers the defined sources, integrates them and presents the consolidated results. Using a mash-up strategy and an information retrieval model, the results are ordered using a relevance score, calculated from the contents of the collected documents.

UFP deals with many developing challenges: (1) the information sources may include non-structured information from diverse origins and formats. This is tackled using a big data approach, considering the information volume and heterogeneity. The information model is flexible and scalable enough to allow integration of new sources and application features; (2) the freshest news and stock prices recollection is considered, giving the user meaningful and updated information without affecting the application performance. New content is collected daily and each content piece is analyzed and transformed for later indexing and storage; (3) news and comments are semantically analyzed considering the characteristic of social media information.

UFP works with three categories of information in two languages: (1) Stock exchanges sources, mainly from the Colombian Stock Exchange, in Spanish, and the New York Stock Exchange, in English. They provide a daily report of all company prices and variations in the market; (2) newspapers feeds from different places, like Portafolio (Colombia), The Economist (England), The Wall Street Journal and The New York Times (USA); (3) social networks, mainly Facebook and Twiter.

[1] Online financial specialized newspapers examples: `www.portafolio.com, www.economist.com, www.nytimes.com, online.wsj.com`

UFP process the information in three stages: First, the system must be configured with the initial interesting data sources and the financial model is established by a domain expert. Next, the defined sources are crawled; the gathered information is classified and indexed. Finally, the user can monitor his relevant information associated to his portfolio using a Web interface (Figure 1), than shows the integrated content, including stock prices, sentiment polarity and relevant news.



Una patadita de la buena suerte a Millonarios

Blog de usuario: La mesa de dinero Leía una columna de opinión de Juan Carlos Ortiz, un joven pero experimentado ex corredor de bolsa, de quién se dice es el hombre fuerte detrás de los grupos financieros Proyectar Valores e Interbolsa, retirado de la extinta Bolsa de Bogotá, y quien casi que de una manera inocente lanzó ayer la siguiente pregunta: "Y qué pasa si Millonarios va a la bolsa? leer m�s

Banco do Brasil aterrizar� este a�o en Colombia

Est� interesado en prestar sus servicios a las compa��as brasile�as que est�n ohar�n presencia en el pa�s. La llegada de Banco do Brasil a Colombia será una realidad este año, a través de la compra de una entidad del sistema financiero local o por medio de la apertura de una oficina de representación, que prestará sus servicios a las compañías brasileñas que están o harán presencia en el país. Así lo anunció el presidente de esta entidad, Carlos Massaru Takana-shi, quien llegó a Bogotá a anunciar una alianza con la Sociedad Administradora de Inversión del Grupo Interbolsa. leer m�s

InterBolsa expande su portafolio y ser� una fiduciaria

La administradora de inversiones del Grupo tramita la conversi�n a fiduciaria. Espera duplicar sus ingresos. Las inversiones en activos diferentes a lo que tradicionalmente ofrece el mercado de capitales será una de las apuestas del Grupo InterBolsa para los próximos meses. La construcción de un centro comercial en Santa Marta, la originación de hipotecas, la creación de un fondo de pensiones voluntarias y hasta las microfinanzas, son algunas de las iniciativas que ya están en trámite o que tienen entre los planes de mediano plazo. leer

EC - NYQ $58.71
Ecopetrol S.A. American Deposit

- Ecopetrol S.A. (EC): Zacks Rank Buy
  Zacks (Wed 1:00am)
- Top World Stocks Hit 50-Day Lines On Economic Worries

Fig. 1. UFP Web User Interface

3.2 Application Architecture

UFP follows the three-tier architecture for Web applications, designed to favor performance and flexibility (Figure 2). Performance is improved using in-memory indexing and separating the component that extracts data from the user end application.

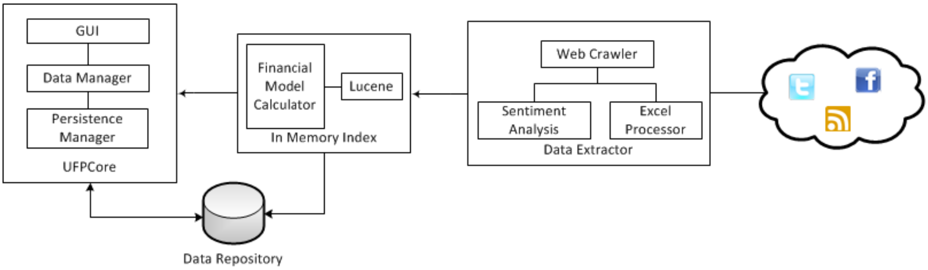


Fig. 2. UFP Components

UFP has three main components: (1) The UFPCore component consists of a GUI layer; a data manager for managing which information must be displayed to which user, according to his portfolio, and the Persistence Manager to query the data repository; (2) The Data Extractor component is composed by a set of Web Crawlers, one for each information source type: one for news feeds, another for the Colombian

Stock Exchange and one for the New York Stock Exchange. This component also handles the social media APIs, mainly Facebook API and Twitter API. The Facebook search is made using the enterprise profiles to obtain comments on the company's wall. Tweets are obtained using enterprise names and delivered to the Sentiment Analysis component to obtain its polarity. The Excel Processor allows the extraction of interesting alphanumeric data provided by the stock sources. (3) The information retrieval model is implemented in the In-Memory Index component, using the Financial Model Calculator to calculate the relevance score for each document obtained by the Data Extractor. This component takes advantage of Apache Lucene [14] to do a full text search and to maintain an in-memory index that maps the enterprise to its relevant documents. The indexing library allows to deal with the documents syntactic heterogeneity. However, a stemming algorithm can be implemented to improve the results and enable a better syntactic understanding of each document.

The gathered information nature leads us to use a NoSQL repository, which offers high scalability, high availability and good response time for queries involving reading Big Data volumes.

3.3 Data Model

The designed data model follows a wide column store type (Figure 3), modeling the source document structure heterogeneity by a set of predefined columns, and new columns may be defined as needed in an extendible way. A portfolio element is the arrangement of user assets. It is important that this model provides a mechanism to find relationships between this element and the stored documents.

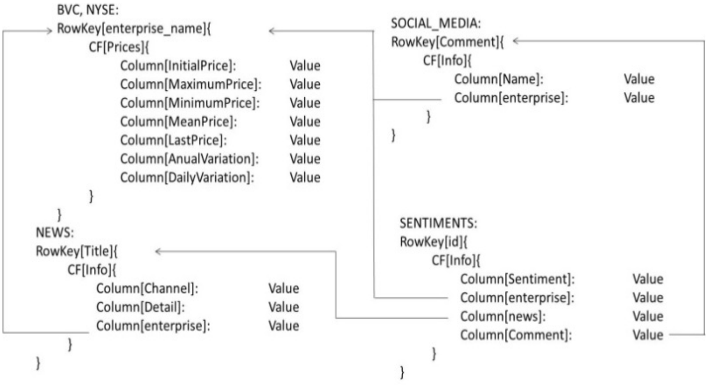
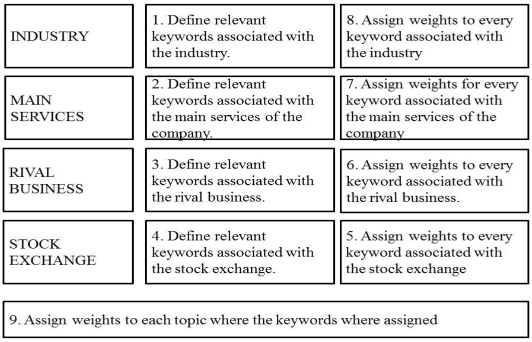


Fig. 3. UFP Data Model

A standard notation or domain specific language for modeling a Column Store data model was not found. The notation in Figure 3 is as follows: Each word in capital letter is the name of a table which has a row key represented in square brackets. Each table can be composed by column families (CF). Each column family has a set of columns represented by curly brackets. Each element in square bracket corresponds to the name of the data element.

3.4 Semantic Table

For each UFP asset, the information retrieval model uses an Asset Semantic Table that associates weights to relevant keywords in the financial portfolio domain. An Asset Semantic Table includes the considered topics, described by a set of weighted keywords, and their relative weight in the system. Each Asset Semantic Table is described in a first stage of UFP operation. For each asset, a domain expert assigns keywords and weights following a nine steps process, as shown in Figure 4.



INDUSTRY	1. Define relevant keywords associated with the industry.	8. Assign weights to every keyword associated with the industry
MAIN SERVICES	2. Define relevant keywords associated with the main services of the company.	7. Assign weights for every keyword associated with the main services of the company
RIVAL BUSINESS	3. Define relevant keywords associated with the rival business.	6. Assign weights to every keyword associated with the rival business.
STOCK EXCHANGE	4. Define relevant keywords associated with the stock exchange.	5. Assign weights to every keyword associated with the stock exchange

9. Assign weights to each topic where the keywords where assigned

Fig. 4. Asset Semantic Table Definition Process

The quality of the model depends on the precision of the weights assigned to the keywords. The domain expert must know the industry, the main services, the rival business and all the activities around the assets that compose the portfolios.

Once all the keywords and weights have been assigned, the Semantic Table is created (eq. 1). The Asset Semantic Table is defined as set of Topics. The UFP Topics are taken from Figure 4. A Topic is defined as:

$$topic = \left[w_{topic}, \left\{kw_{topic_i}, rkw_{topic_i}, tkw_{topic_i}\right\}\right], \left\{kw_{topic_i}, rkw_{topic_i}, tkw_{topic_i}\right\} \neq \emptyset \quad (1)$$

$$\sum\nolimits_{i \in topics} w_i = 1 \quad (2)$$

$$\sum\nolimits_{i \in KW_t} rkwt_i = 1, \forall t \in topics \quad (3)$$

$$tkw_{t_i} = w_t * rkw_{t_i}, \forall t \in topics, \forall i \in KW_t \quad (4)$$

where w_{topic} corresponds to the topic weight; kw_{topic_i} is a particular topic keyword; rkw_{topic_i} is the relative weight for the topic keyword; and tkw_{topic_i} is the absolute weight for the topic keyword. When UFP retrieves the information of a specific asset, the semantic table is used to search not only the asset name, but also the associated keywords. The weights are used to calculate the final document score given an asset. These equations (eq. 2, 3, 4) define the relative importance of each topic keyword as well as the importance of an asset topic.

3.5 Vector-Space Model

Apache Lucene is a high-performance, full-featured text search engine library that uses Boolean Model [15] to approve documents, and Vector-Space Model [15] to calculate document scores. Because of the document characteristics managed by UFP and the Asset Semantic Table definition, a Vector-Space model approach is selected, extending the Lucene text search engine with specific financial domain factors to generate an overall score.

$$UFPscore_{(q,d)} = C_{(q,d)} \cdot norm_{(q)} \cdot sumt_{(q,d)} \tag{5}$$

$$norm_q = \frac{1}{\sqrt{\sum_{t \epsilon q} (idf_{(t)} \cdot tkw_{(t)})^2}} \tag{6}$$

$$sumt_{(q,d)} = \sum_{t \epsilon q} (tf_{(t,d)} \cdot idf_{(t)}{}^2 \cdot tkw_{(t)}) \tag{7}$$

In the score formula (eq. 5) of the Vector-Space Model $C_{(q,d)}$, represents how many keywords are found in the document; $norm_{(q)}$ is a normalizing factor making comparable scores between queries, using the inverse document frequency sum for all the terms in order to convert the final score to a normal form (eq. 6); $sumt_{(q,d)}$ is the contribution of each keyword to the query score. It is based in the frequency of a keyword in the document, the inverse document frequency where rarer keywords get higher scores and the total weight of the keyword (eq. 7).

3.6 Sentiment Analysis of Market Perception

One of the more important contributions of UFP is the processing of social media feeds that denotes the market perception of recent news. The amount of information generated by social networks increases rapidly and leads to unread or useless information. In order to avoid the loss of relevant feeds, UFP integrates a sentiment analysis tool, qualifying them as positive, neutral or negative. UFP presents to the user a brief summary of all these social media feeds, with the total of positive, neutral and negative perceptions for a particular asset, giving him a general idea of the gathered information. Besides, if more specific information is needed, it is possible to look at any individual relevant feed with the particular classification given by the emotion engine. UFP selected Synesketch [16] [17] as the sentiment analysis technology because it is a free open-source engine for textual emotion recognition, and allows multiple language configurations for domain sentiment inference, which is an interesting feature for the multi-language source integration.

4 Implementation, Test and Validation

A fully functional Web prototype was implemented in a Debian 6 cluster, using Java and Glassfish. The user's portfolio composition, an overview of the social media polarity and latest news related to his portfolio are displayed and can be filtered for obtaining the corresponding relevant information. Apache Hbase [18], a Column-Store database technology, was successfully used to handle sparse data. The Data Extractor component recollected data for 52 days from all the selected sources; the Stock Exchange provided data only on 38 working days during this period. 242 different new feed sources, 29919 financial news and 3152 elements from the social media (mainly from Twitter) were collected.

The model performance evaluation considers the relationship between the returned documents and the portfolio composition. The news and the social media content presented to a user are effectively related to his portfolio, either because it is related directly to a company, or because it has a relationship through the business domain. Table 1 presents the query evaluation for an example user portfolio, where the user is interested in four Latin-American regional enterprises in energy, bank, petroleum and financial services: Odinsa, Davivienda, Ecopetrol and Interbolsa, respectively. The evaluation was made establishing a ground truth of a hundred documents, where each one was classified manually. Each document was rated 0 (if the document is not relevant for the user portfolio) or 1 (the document is relevant for the user portfolio).

Table 1. Retrieval results for news content

Company	*Precision*	*Recall*
Odinsa	70%	84%
Interbolsa	28,5%	44%
Davivienda	75%	76%
Ecopetrol	68,9%	88%

Our results are comparable with the Hermes framework and with YourNews works (see Section 2). In general, they achieved a higher precision but a lower recall than our results, without using a specific domain ontology.

The queries are executed using the gathered news database and the Asset Semantic Tables defined for each company. The results concerning news contents have a good recall and precision. It is worth to explain the results obtained for Interbolsa: Given that this company has a very wide range of associated topics, the semantic table for this company is very sparse and has a negative impact on the model performance.

Table 2 presents relevance results of queries considering the social media content. These results are not as good as the precedent, mainly attributed to two factors. (1) Social media content is very informal, and slang expressions are frequent. (2) In the original classification model of Synesketch only common language sentiment expressions were considered, leading to poor polarity identification. In order to improve these results, the sentiment analysis tool must consider local domain customization. The evaluation of Synesketch in [19] shows that its performance is not the best which has a negative effect on the UFP model. LingPipe [20] would be a better solution as shown in [21], even if it requires a previous training stage. In fact, LingPipe was not used for two reasons. First, the application deals with multi-language documents and LingPipe must be trained in each language to have good performance. Second, given the developing times of the project, Synesketch is preferred.

Table 2. Retrieval Results for Sentiment Analysis on Social Media Content

Company	*Precision Positive Elements*	*Precision Negative Elements*	*Recall Positive Elements*	*Recall Negative Elements*
Odinsa	33%	39%	30%	40%
Interbolsa	67%	33%	20%	60%
Davivienda	41%	36%	30%	20%
Ecopetrol	39%	34%	50%	40%

5 Conclusion and Future Work

The design and development of Unified Financial Portfolio, an application that successfully integrates and retrieves non-structured, heterogeneous and domain-specific contents from more than 240 information sources, having multi-language, international and regional coverage, has been presented. It uses a specific semantic model for domain information integration based on the vector-space model in order to monitor the user financial portfolio. The information delivery is adapted to the user financial information needs, using access by content as the main criterion for classification and information retrieval, both for Web and structured public data. The content is analyzed either to recognize financial features or to use sentiment analysis on social media sources. Finally, the collected content is stored in a NoSQL repository that was successfully integrated within the application.

The proposed architecture can be easily used in other contexts, other than financial information. The semantic table can be defined with any domain specific terms, the syntactic text analysis can be trained as well with the corresponding terms and the interesting content sources can be easily included at running time. Then, the proposed model can be adapted to other domains. The final user interface, of course, must be redesigned.

Future work and challenges are identified from the obtained results. The sentiment analysis can be enhanced using other tools allowing better polarity recognition, as well as local or regional sentiment expressions. Dynamic integration of new assets once the system is up and running can avoid the manual definition of the Asset Semantic Table, used to weight the queries against the several information sources. Dynamic ordering of the obtained retrieval results, considering user portfolio composition, or features like financial risk measures could be included. The growth of the document database is very important, not only using the same sources of information but also integrating new sources dynamically. The last two proposed modifications could take advantage of the user knowledge, interest and interaction with the application. Finally, to enable the previous modifications and to extend the information retrieval model, financial domain ontology and syntactical stemming can be integrated in the system, allowing a refinement in search queries and using semantic relationships based on already defined keywords synonyms.

References

1. Google Finance, http://www.google.com/finance
2. Yahoo! Finance, http://finance.yahoo.com/
3. Bloomberg, http://www.bloomberg.com/
4. O'Reilly, http://oreilly.com/web2/archive/what-is-web-20.html
5. Baeza-Yates, R., Ribeiro-Neto, B., Maarek, Y.: Web Retrieval. In: Modern Information Retrieval the Concepts and Technology behind Search, pp. 449–517. Addison-Wesley (2011)
6. Baeza-Yates, R., Ribeiro-Neto, B., Castillo, C.: Web Crawling. In: Modern Information Retrieval the Concepts and Technology Behind Search, pp. 519–548. Addison-Wesley (2011)
7. Shah, V.: Stoocker, http://www.stoocker.com
8. Schouten, K., Ruijgrok, P., Borsje, J., Frasincar, F., Levering, L., Hogenboom, F.: A semantic web-based approach for personalizing news. In: The 2010 ACM Symposium on Applied Computing (SAC 2010), pp. 854-861 (2010)
9. Mast, L., Micu, A., Frasincar, F., Milea, V., Kaymak, U.: StockWatcher - A Semantic Web Application for Custom Selection of Financial News. In: Second Knowledge Management in Organizations Conference (KMO 2007), pp. 121–126 (2007)
10. Ahn, J.-W., Brusilovsky, P., Grady, J., He, D., Syn, S.: Open user profiles for adaptive news systems: help or harm. In: The 16th International Conference on World Wide Web, pp.11–20 (2007)
11. Java, A., Finin, T., Nirenburg, S.: SemNews: a semantic news framework. In: The 21st National Conference on Artificial Intelligence (AAAI 2006), pp. 1939–1940 (2006)
12. Castells, P., Fernandez, M., Vallet, D.: An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. IEEE Transactions on Knowledge and Data Engineering 19(2), 261–272 (2007)
13. Voulodimos, A., Patrikakis, C.: Using Personalized Mashups for Mobile Location Based Services. In: Wireless Communications and Mobile Computing Conference (2008)
14. The Apache Software Foundation: Lucene, http://lucene.apache.org/
15. Croft, B., Metzler, D., Strochman, T.: Search Engine-Information Retrieval in Practice. Pearson Education (2010)
16. Krcadinac, U.: Synesketch, http://synesketch.krcadinac.com/
17. Krcadinac, U., http://www.krcadinac.com/papers/synesketch_paper_in_serbian.pdf
18. The Apache Software Foundation: Apache Hbase, http://hbase.apache.org/
19. Moshfeghi, Y.: Role of Emotion in Information Retrieval. University of Glasgow, Glasgow Theses Service, http://theses.gla.ac.uk/3118/
20. Alias-i, Inc.: LingPipe, http://alias-i.com/lingpipe/
21. Pérez-Granados, D., Lozano-Garzón, C., López-Urueña, A., Jiménez-Guarín, C.: Sentiment Analysis in Colombian Online Newspaper Comments. In: Gaol, F. (ed.) Recent Progress in Data Engineering and Internet Technology. LNEE, vol. 157, pp. 113–119. Springer, Heidelberg (2012)

Financial Events Recognition in Web News for Algorithmic Trading

Frederik Hogenboom

Erasmus University Rotterdam,
P.O. Box 1738, NL-3000 DR,
Rotterdam, The Netherlands
fhogenboom@ese.eur.nl

Abstract. Due to its high productivity at relatively low costs, algorithmic trading has become increasingly popular over the last few years. As news can improve the returns generated by algorithmic trading, there is a growing need to use online news information in algorithmic trading in order to react real-time to market events. The biggest challenge is to automate the recognition of financial events from Web news items as an important input next to stock prices for algorithmic trading. In this position paper, we propose a multi-disciplinary approach to financial events recognition in news for algorithmic trading called FERNAT, using techniques from finance, text mining, artificial intelligence, and the Semantic Web.

1 Introduction

Recently, financial markets have experienced a shift from the traditional way of trading using human brokers to the use of computer programs and algorithms for trading, i.e., algorithmic trading. Trading algorithms implemented in business tools have proven to be more efficient than conventional approaches for trading, as they provide for lower latency, larger volume, and higher market coverage degree. During the last years, it has been acknowledged the need to use Web news information in algorithmic trading in order to react real-time to market events and to enable better decision making. Responding to market events only a few milliseconds faster than the competition could mean better prices and therewith improved profitability for traders. The biggest challenge is to allow machines to identify and use the news information that is relevant for technical trading timely and accurately.

Financial markets are extremely sensitive to breaking news [22]. Financial events – phenomena that are captured in keywords pointing to specific (complex) concepts related to money and risk – like mergers and acquisitions, stock splits, dividend announcements, etc., play a crucial role in the daily decisions taken by brokers, where brokers can be of human or machine nature. Algorithmic trading enables machines to read and understand news faster than the human eye can scan them, hence allowing one to deal with larger volumes of emerging online news, and making thus better informed decisions.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 368–377, 2012.

The Semantic Web provides the right technologies to unambiguously identify or "tag" the semantic information in news items and represent it in a machine-understandable form. Having this information in machine-understandable form enables computers to reason and act as we humans would do. Realizing the potential the Semantic Web has to offer in making the news information semantically available, large news companies like Reuters and Dow Jones started to provide product services that offer tagged news items to be used for algorithmic trading [27].

The current annotations provided by the above vendors are *coarse-grained*, as they supply general information about the type of information available in news items, as for example company, topic, industry, etc., satisfying thus to a limited extent the information need in financial markets. For algorithmic trading, a *fine-grained* annotation [8] that allows the identification of financial events as acquisitions, stock splits, dividend announcements, etc., is needed. Additionally, most annotations are merely based on article titles instead of contents, and financial events (if any) are not linked to ontologies (hence making reasoning and knowledge inference difficult).

To our knowledge the semi-automatic recognition of financial events as a support tool for algorithmic trading has not been thoroughly investigated in previous work. Several innovative aspects play a key role here: defining a *financial ontology* for algorithmic trading, using *lexico-semantic rules* for identifying financial events in news, applying *ontology update rules* based on the previously extracted information, and employing the financial events to improve the returns generated by *trading algorithms*. In recent work, we have focused on the first two aspects. Our main contribution in the field of news analysis is the Hermes framework [12], which makes use of Natural Language Processing (NLP) techniques and Semantic Web technologies for news personalization. Additionally, we researched financial ontologies and financial event detection pipelines [4,14] and we have introduced a lexico-semantic pattern language for Hermes [15] which is able to extract financial events from text using a financial ontology.

In light of our existing work, this paper presents the Financial Events Recognition in News for Algorithmic Trading (FERNAT) framework, which aims to automate the identification of financial events in emerging news and to apply these events to algorithmic trading. Not only does the proposed framework make use of an NLP pipeline, a financial ontology, and lexico-semantic patterns for event extraction resulting from earlier work, but it also implements a feedback loop using ontology update rules. Additionally, the discovered events are used for financial applications for risk analysis or algorithmic trading.

2 Related Work

This section discusses related work with respect to information extraction frameworks, and compares these with our proposed FERNAT framework. Additionally, we discuss work on trading in the financial markets and algorithmic trading.

2.1 Information Extraction

For the information extraction methods, we distinguish between general-purpose text processing pipelines and news-based processing frameworks. Examples of general-purpose text processing pipelines are A Nearly New Information Extraction System (ANNIE) [6] and Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations (CAFETIERE) [3]. For news-based processing frameworks we identify PlanetOnto [7] and SemNews [16].

Both ANNIE and CAFETIERE are able to cope with the domain semantics to a limited extent. For gazetteering, their pipelines use a list of words for which the semantics are not defined. Also, the information extraction rules are based on lexico-semantic patterns that apply only to very concrete situations. These rules are written in JAPE [6] using Java, a low-level format which makes rules development and maintenance rather tedious. The MUlti-Source Entity finder (MUSE) [21] uses the ANNIE pipeline for named entity recognition going through the rather difficult process of defining JAPE rules for information extraction.

There are a number of tools for Ontology-Based Information Extraction (OBIE) that have adapted the ANNIE pipeline to be used in combination with ontologies. Examples of such tools are the Ontology-based Corpus Annotation Tool (OCAT) [20] that has been used for annotating documents for business intelligence purposes, and the Knowledge and Information Systems Management (KIM) platform [26], a generic annotation and document management system. While these tools benefit from the information stored in ontologies for knowledge acquisition, due to the direct use of JAPE rules, they fail to deliver an easy-to-use, high-level language for information extraction rules specification.

Differently than ANNIE, CAFETIERE provides high-level information extraction rules which makes it easier to write and update rules. Although the extraction rules are defined at lexico-semantic level, CAFETIERE does not employ ontologies (knowledge bases) that are Semantic Web-based. For this, CAFETIERE uses a specific representation, i.e., Narrative Knowledge Representation Language (NKRL), a knowledge representation language which is defined before the Semantic Web era. With the advent of the Semantic Web, we believe that both gazetteering and lexico-semantic rules can benefit from an ontology-based approach based on standards and proven tool support. Also, both ANNIE and CAFETIERE do not update their knowledge bases with extracted information that is possibly helpful in the next information extraction run.

PlanetOnto represents an integral suite of tools used to create, deliver, and query internal newsletters of the Knowledge Media Institute (KMi). Similar to the approach proposed here, domain ontologies are used for identifying events in news items. While we aim at semi-automatic information extraction from news items, PlanetOnto uses a manual procedure for identifying information in news items. SemNews on the other hand uses a domain-independent ontology for semi-automatically translating Web pages and RSS feeds onto meaningful representations given as OWL facts. For this purpose it uses OntoSem [25], an NLP tool which performs lexical, syntactic, and semantic analysis of text. OntoSem has a specific frame-based language for representing the ontology and an onomasticon for storing proper names. In our work both the input ontology and the facts extracted from news items are to be represented in OWL. Also, our approach proposes to use, instead of an onomasticon, a semantic lexicon, a richer knowledge base that can better support the semantic analysis of text.

Many of the current approaches for automating information extraction from text use rules based on lexico-syntactic patterns. As these rules do not take into account the semantics of the different constructs involved in a pattern we do find such an approach limited in expressiveness. In our previous work [15] we aim at exploiting lexico-semantic patterns, which remove some of the ambiguity inherent to the lexico-syntactic rules. In addition, the proposed rules provide a higher abstraction level than lexico-syntactic rules, making rule development and maintenance easier.

2.2 Financial Markets

Financial markets are strongly dependent on information, and thus also on emerging news messages. Traders – whether they are technical or fundamental traders – use information in their decisions on selling and buying stocks, thus influencing the financial market. Processing and interpreting relevant information in a timely manner can be of crucial importance for the profitability of trading activities. Predicting the future course of stocks within a financial market is hard, which led to the development of theories on stock prediction, such as the random walk theory [9] and the efficient market hypothesis [10], that both recognize the influence of available information on the market.

As shown in the previous section, an extensive body of literature is available on processing text to a machine-understandable format. Also, a lot of research has been done for the prediction of market reactions to news (see [23] for an extensive survey). Many existing approaches aim to forecast price trends based on emerging news and mainly employ statistical text mining approaches to classify financial events (e.g., positive or negative). Price trends based on news messages can be used in automated trading algorithms. Examples of these algorithms are the Penn-Lehman Automated Trader (PLAT) [17], the Artificial Stock Trading Agent (ASTA) [13], and genetic algorithms-based financial trading rules [1].

Algorithmic trading encompasses the use of computer programs for trading purposes, which is of interest to traders as this greatly enhances trading speed, and thus increases profit expectations. Algorithms are employed for instance for correlation analyses and the identification of opportunities and risks. These algorithms are based on inputs such as statistics on the financial market, but also price trends. These price trends can be calculated based on both historical and real-time market data. However, real-time market data as for example news information is often inaccurate or too coarse to be of great value. Thus, improving processing speed and accuracy of real-time information would be beneficial for algorithmic trading.

3 FERNAT Framework

The Financial Events Recognition in News for Algorithmic Trading (FERNAT) framework proposes a pipeline to extract financial evnets from news to be exploited in algorithmic trading. First, news messages (i.e., written text in natural language that originate from RSS sources) are parsed to tokens. These tokens are then used to match patterns that identify (extract) financial events. Then, these events are used in decision making, i.e., trading in financial markets. This section continues with discussing the proposed model in more detail.

3.1 Processing Pipeline

The first part of our framework, i.e., news extraction trough a processing pipeline, is depicted in Fig. 1. The pipeline contains two parts: the lexico-syntactic analysis and the semantic analysis. The cornerstone of the pipeline is a domain ontology for financial events and their related facts. This information defines the expert view on the financial world at a certain moment in time useful for algorithmic trading. The concepts defined in the ontology are anchored to the synsets defined in a semantic lexicon, if such synsets exist. The purpose of the semantic lexicon is twofold: to help define the meaning of the domain concepts and to have access to more lexical representations (lexons) for the ontology concepts.

The lexico-syntactic analysis comprises the following processing units: text tokenizer, sentence splitter, Part-of-Speech (POS) tagger, morphological analyzer, and lexon recognizer. The text tokenizer recognizes the basic text units (tokens) such as words and punctuation. Then, the sentence splitter identifies the sentences present in the news items. After that, the morphological analyzer determines the lemma associated with each word in the text. The POS tagger associates to each word its grammatical type (e.g., noun, verb, pronoun, etc.). The lexon recognizer identifies using gazetteers lexical representations of concepts from both the domain ontology and the semantic lexicon present in news items. The lexons found outside the ontology are useful for defining the contextual meaning of a sentence, a feature exploited in the next processing unit.

The semantic analysis consists of the following processing units: lexon disambiguator, event recognizer, event decorator, ontology instantiator, and ontology updater. The lexon disambiguator uses word sense disambiguation techniques, as for example Structural Semantic Interconnections (SSI) [24], for computing the senses of the found lexons. The lexons which correspond to the financial events stored in the ontology are used for building event instances in the event recognizer. For example, the word "acquire" in the sentence "Google acquires Appjet for Word Processing Collaboration and Teracent to Beef Up Display Ad" is recognized as instance of the ontology referred to by prefix `kb`, i.e., `kb:BuyEvent`.

The event decorator uses *lexico-semantic patterns* to mine facts relevant for event description. An illustration of such a rule is a pattern that mines texts for company acquisitions, i.e.,

```
$sub:=[kb:Company] $prd:=kb:BuyEvent $obj:=([kb:Company])+
```

where `$sub`, `$prd`, and `$obj` are variables representing a buyer, buy event, and buyee, respectively. Furthermore, `kb:BuyEvent` and `[kb:Company]` are an instance and a class from the ontology, and `+` is the repetition operator. Based on this rule (when applied on our earlier example), `$prd` represents the previously

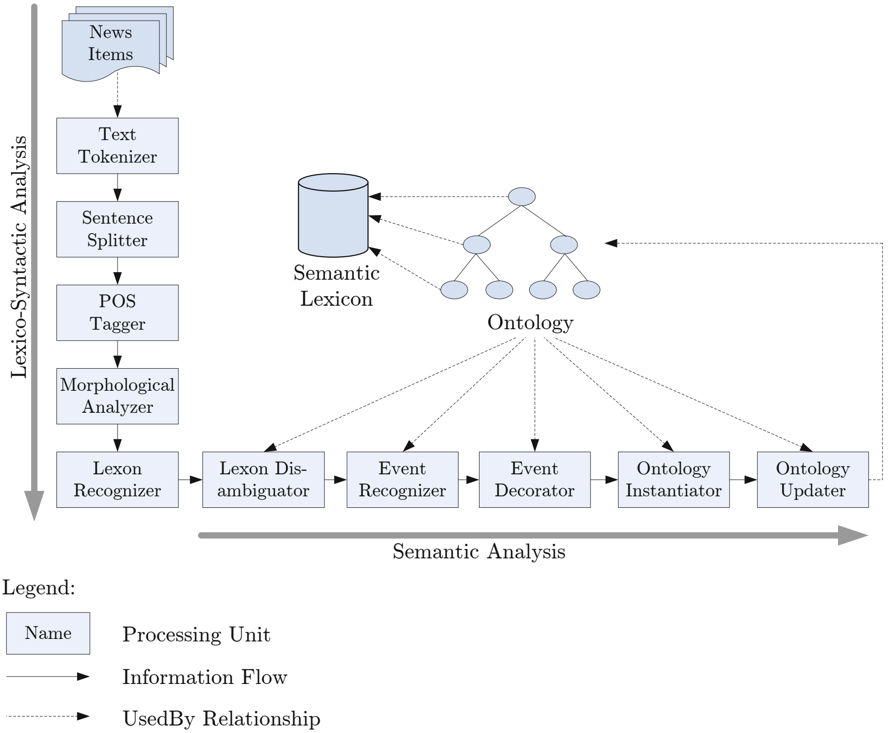


Fig. 1. The FERNAT processing pipeline

discovered event, `$sub` is assigned to the ontology instance "Google", and `$obj` is assigned an array with the ontology instances "Appjet" and "Teracent", respectively. The lexico-semantic patterns leverage existing lexico-syntactic patterns to a higher abstraction level by using ontology concepts in the pattern construction. Also, in this processing unit, the time associated to an event instance is determined. The discovered events and their related facts need to be manually validated before the next step can proceed in order to prevent erroneous updates to cascade through the ontology, possibly causing incorrect trading decisions when used in algorithmic trading. In the ontology instantiator, the events and their associated information are inserted in the ontology.

In the last processing unit, the ontology updater uses *update rules* for implementing the effects of the discovered events in the domain ontology. An illustration of such a rule is

```
$prd:=kb:BuyEvent($sub:=[kb:Company], $obj:=[kb:Company])
-> DELETE    $sub kb:hasCompetitor $obj
   CONSTRUCT $sub kb:owns $obj
```

where `kb:hasCompetitor` and `kb:owns` are ontology relationships. In our running example, knowing that Google buys Appjet and Terracent would imply that Google and the other two companies are not anymore in the competitor relation and are now parts of the same company. While this example is about learning new relations, ontology update rules can also be used for learning new instances (e.g., a company which is not yet present in the ontology). The purpose of the ontology updates is twofold: extracting more information from subsequent news items, and providing more information in the ontology useful for algorithmic trading.

The innovation of the proposed approach stems from several issues. First, it proposes a methodology for extracting in a semi-automatic manner financial events from news items. The recognized financial events are to be used as additional input next to stock prices for trading algorithms. Second, it investigates the use of ontologies and semantic lexicons for information extraction at multiple methodological levels: domain modeling, gazetteering, word sense disambiguation, information extraction pattern construction, knowledge base update rule building, and result delivery. The envisaged ontology gazetteer is expected to go beyond state-of-the-art ontology gazetteers by allowing the automatic generation of gazetteer's lists from the ontology content. Third, it proposes the use of lexico-semantic patterns, a generalization of lexico-syntactic patterns, which makes easier the pattern development and maintenance. Last, but not least, by using update rules it implements the changes to the financial world implied by the discovered events in the domain ontology.

The implementation of the proposed methodology requires a large number of technologies like text mining tools such as GATE components, Semantic Web languages as RDF and OWL, and semantic lexicons like WordNet [18]. As most of these technologies are supported by Java libraries we develop an implementation based on the Java programming language. As input we use RSS news feeds originating from different online sources, e.g., Reuters, BBC, NYT, etc. Most of the components depicted in Fig. 1 can be implemented by reusing existing implementations. For example, the text tokenizer and sentence splitter can be implemented using ANNIE components [6], while the POS tagger can be implemented by means of the Stanford POS tagger [28]. Subsequently, the MIT Java Wordnet Interface (JWI) [19] can be employed for morphological analysis. Additionally, the lexon recognizer, lexon disambiguator, event recognizer, and ontology instantiator can be reused from our earlier work [12,15]. Finally, although we have introduced an ontology update implementation [11], we can extend this with the more expressive update language proposed in this paper.

Performance-wise, we aim to be able to rapidly and correctly identify most of the financial events present in news as required by an algorithmic trading setup. For this purpose we aim for state-of-the-art performance, i.e., precision and recall of 70-80% and sub-second performance for news processing time.

3.2 Algorithmic Trading

For the second part of the framework we investigate the use of the extracted financial events for improving the returns of technical trading rules. More precisely, we plan to associate stock price impact factors to financial events that quantify what is the effect of a financial event on a stock price. By aggregating the stock price impacts of the events found in a news item, we can determine a trading signal (e.g., buy, hold, or sell) given by the news item. Then, by combining signals generated by news items with signals obtained through technical trading, we can provide for an aggregated signal that better reflects the current situation than by using technical trading alone.

For this purpose we plan to extend a genetic programming approach which generates high performing technical trading rules. As other approaches are mostly based on ad-hoc specifications of trading rules [2,5], by using an evolutionary algorithm we avoid the danger of ex post selection, and are able to generate rules that are in a sense optimal. Our choice for genetic programming is also motivated by the easy extension of the genetic programming solution for our current purpose, by adding news-based signals as leaves in the trading rule tree, in addition to the technical trading rules. As a last step, we will show that most of the generated (optimal) technical trading rules do make use of news and thus provide for better returns than the ones which do not make use of the news component.

High frequency trading without accounting for news is undoubtedly faster than the framework proposed in this paper, yet it does not take into account an updated knowledge base with the latest facts, generating trading decisions less informed and accurate. Lags between the publication of news and the reaction of the stock market could be substantial enough to cover for the increase in processing time caused by the usage of rather heavy Semantic Web technologies. Furthermore, one could also separate the computationally intensive event recognition tasks from algorithmic trading. This way, the knowledge base containing market facts is updated once news is processed. Trading algorithms are to be run in separate processes and make use of the (regularly updated) knowledge base, reducing the reaction time on the financial markets. Being able to reason with the financial information stored in the ontology will provide for an increased support for trading decisions.

4 Conclusions

In this position paper we have proposed the FERNAT framework for financial event recognition in news, which encompasses a news processing pipeline of which the outputs are applied to algorithmic trading. The framework builds partially on earlier work for its NLP tasks and makes use of our developed financial ontology and lexico-semantic event extraction pattern language. For ontology updating we have briefly touched upon a proposal for an update language which needs to be implemented in future work. Additional further work is related to the proposed application for algorithmic trading. We envision a genetic programming approach that generates high-performing technical trading rules through the usage of stock price impact factors associated to financial events discovered by the proposed news processing pipeline.

Acknowledgement. The author is sponsored by the NWO Physical Sciences Free Competition project 612.001.009: Financial Events Recognition in News for Algorithmic Trading (FERNAT) and the Dutch national program COMMIT.

References

1. Allen, F., Karjalainen, R.: Using Genetic Algorithms to Find Technical Trading Rules. Journal of Financial Economics 51(2), 245–271 (1999)
2. Bessembinder, H., Chan, K.: The profitability of Technical Trading Rules in the Asian Stock Markets. Pacific-Basin Finance Journal 3(2–3), 257–284 (1995)
3. Black, W.J.: M^{c}Naught, J., Vasilakopoulos, A., Zervanou, K., Theodoulidis, B., Rinaldi, F.: CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations. Technical Report TR–U4.3.1, Department of Computation, UMIST, Manchester (2005), http://www.nactem.ac.uk/files/phatfile/cafetiere-report.pdf
4. Borsje, J., Hogenboom, F., Frasincar, F.: Semi-Automatic Financial Events Discovery Based on Lexico-Semantic Patterns. International Journal of Web Engineering and Technology 6(2), 115–140 (2010)
5. Brock, W.A., Lakonishok, J., LeBaron, B.: Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. Journal of Finance 47(5), 1731–1764 (1992)
6. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), pp. 168–175. Association for Computational Linguistics (2002)
7. Domingue, J., Motta, E.: PlanetOnto: From News Publishing to Integrated Knowledge Management Support. IEEE Intelligent Systems 15(3), 26–32 (2000)
8. Drury, B., Almeida, J.J.: Identification of Fine Grained Feature Based Event and Sentiment Phrases from Business News Stories. In: Akerkar, R. (ed.) International Conference on Web Intelligence, Mining and Semantics (WIMS 2011). ACM (2011)
9. Fama, E.F.: The Behavior of Stock-Market Prices. Journal of Business 38(1), 34–105 (1965)
10. Fama, E.F.: Efficient Capital Markets: A Review of Theory and Empirical Work. Journal of Finance 25(2), 383–417 (1970)
11. Frasincar, F., Borsje, J., Hogenboom, F.: Personalizing News Services Using Semantic Web Technologies. In: E-Business Applications for Product Development and Competitive Growth: Emerging Technologies, pp. 261–289. IGI Global (2011)
12. Frasincar, F., Borsje, J., Levering, L.: A Semantic Web-Based Approach for Building Personalized News Services. International Journal of E-Business Research 5(3), 35–53 (2009)
13. Hellstrom, T., Holmstrom, K.: Parameter Tuning in Trading Algorithms using ASTA. In: 6th International Conference Computational Finance (CF 1999), pp. 343–357. MIT Press (1999)
14. Hogenboom, A., Hogenboom, F., Frasincar, F., Kaymak, U., Schouten, K., van der Meer, O.: Semantics-Based Information Extraction for Detecting Economic Events. Multimedia Tools and Applications, Special Issue on Multimedia Data Annotation and Retrieval using Web 2.0 (to appear, 2012), doi: 10.1007/s11042-012-1122-0

15. IJntema, W., Sangers, J., Hogenboom, F., Frasincar, F.: A Lexico-Semantic Pattern Language for Learning Ontology Instances from Text. Journal of Web Semantics: Science, Services and Agents on the World Wide Web (to appear, 2012), doi: 10.1016/j.websem.2012.01.002
16. Java, A., Finin, T., Nirenburg, S.: Text Understanding Agents and the Semantic Web. In: 39th Hawaii International Conference on Systems Science (HICSS 2006), vol. 3, p. 62b. IEEE Computer Society (2006)
17. Kearns, M.J., Ortiz, L.E.: The Penn-Lehman Automated Trading Project. IEEE Intelligent Systems 18(6), 22–31 (2003)
18. Laboratory, P.C.S.: A Lexical Database for the English Language (WordNet) (2008), http://wordnet.princeton.edu/
19. Finlayson, M.: JWI – the MIT Java Wordnet Interface (2012), http://projects.csail.mit.edu/jwi/
20. Maynard, D., Saggion, H., Yankova, M., Bontcheva, K., Peters, W.: Natural Language Technology for Information Integration in Business Intelligence. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 366–380. Springer, Heidelberg (2007)
21. Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., Wilks, Y.: Architectural Elements of Language Engineering Robustness. Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data 8(1), 257–274 (2002)
22. Mitchell, M.L., Mulherin, J.H.: The Impact of Public Information on the Stock Market. Journal of Finance 49(3), 923–950 (1994)
23. Mittermayer, M.A., Knolmayer, G.F.: Text Mining Systems for Market Response to News: A Survey. Working Paper 184, Institute of Information Systems, University of Bern (2006), http://www2.ie.iwi.unibe.ch/publikationen/berichte/resource/WP-184.pdf
24. Navigli, R., Velardi, P.: Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(7), 1063–1074 (2005)
25. Nirenburg, S., Raskin, V.: Ontological Semantics. MIT Press (2004)
26. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: KIM - A Semantic Platform For Information Extraction and Retrieval. Journal of Natural Language Engineering 10(3-4), 375–392 (2004)
27. Reuters: Reuters NewsScope Archive (2012), http://www2.reuters.com/productinfo/newsscopearchive/
28. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003), pp. 252–259. Association for Computational Linguistics (2003)

Alf-Verifier: An Eclipse Plugin for Verifying Alf/UML Executable Models

Elena Planas[1], David Sanchez-Mendoza[1], Jordi Cabot[2], and Cristina Gómez[3]

[1] Universitat Oberta de Catalunya, Spain
{eplanash,dsanchezmen}@uoc.edu
[2] École des Mines de Nantes - INRIA, France
jordi.cabot@inria.fr
[3] Universitat Politécnica de Catalunya, Spain
cristina@essi.upc.edu

Abstract. In this demonstration we present an Eclipse plugin that implements a lightweight method for verifying fine-grained operations at design time. This tool suffices to check that the execution of the operations (specified in Alf Action Language) is consistent with the integrity constraints defined in the class diagram (specified in UML) and returns a meaningful feedback that helps correcting them otherwise.

1 Introduction

Executable models are models described in sufficient detail so that they can be systematically implemented/executed in the production environment. Executable models play a cornerstone role in the Model-Driven Development (MDD) paradigm, where models are the core artifacts of the development lifecycle and the basis to generate the final software implementation.

Executable models are not a new concept [7] but are now experiencing a comeback. As a relevant example, the OMG has recently published the first version of the "Foundational Subset for Executable UML Models" (fUML) standard [4], an executable subset of the UML that can be used to define, in an operational style, the structural and behavioural semantics of systems. The OMG has also published a beta version of the "Action Language for fUML" (Alf) standard [3], a concrete syntax conforming to the fUML abstract syntax, that provides a textual notation to specify the fine-grained behaviour of systems.

Given the increasing importance of executable models and their impact on the final quality of software systems, the existence of tools to verify the correctness of such models is becoming crucial. Unfortunately, despite the number of research works targetting the verification of behavioural specifications, their computational cost and poor feedback makes them difficult to integrate in current software development processes and tools.

In order to overcome the limitations of the existing tools, in this demonstration we present an Eclipse plugin that implements a lightweight method for verifying executable models. Our tool focuses on the verification of the *Strong Executability*

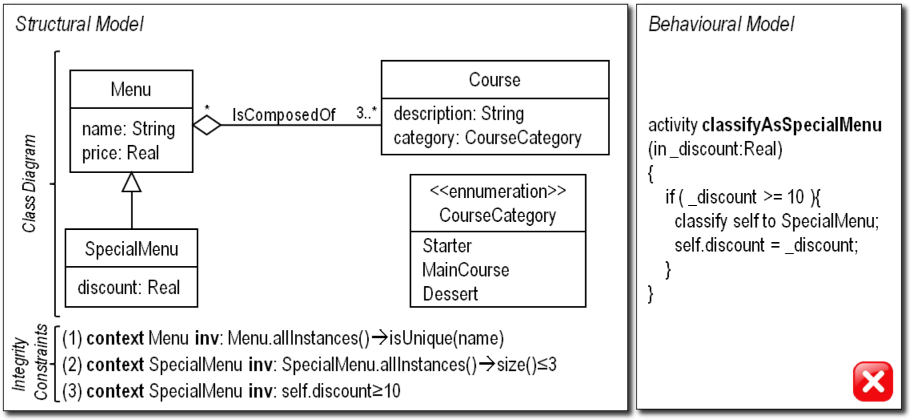
S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 378–382, 2012.

correctness property of operations (attached to domain classes of a UML class diagram) specified by means of Alf Action Language.

To the best of our knowledge, our tool is the first one which deals with verification of Alf specifications.

2 Executability of Operations

We consider that an operation is *Strongly Executable* (SE) if it is always successfully executed, i.e. if every time we execute the operation (whatever values are given as arguments for its parameters), the effect of the actions included in the operation evolves the initial state of the system to a new system state that satisfies all integrity constraints of the structural model.



```
activity classifyAsSpecialMenu
(in _discount:Real)
{
  if ( _discount >= 10 ){
    classify self to SpecialMenu;
    self.discount = _discount;
  }
}
```

Fig. 1. Excerpt of a menus management system executable model

As an example, consider the executable model shown in Fig. 1, meant to model the menus offered by a restaurant. Note that the operation `classifyAsSpecialMenu`, which classifies a menu as special menu, is not SE since after its execution the maximum integrity constraint of class *SpecialMenu* (second constraint) may be violated (in particular, when the system state where the operation is applied contains already three special menus).

In [5] we have published the theoretical background of a lightweight method for verifying the strong executability of Alf operations. This method takes as input an executable model (composed by a UML class diagram and a set of Alf operations) and returns either a positive answer, meaning that the operation is SE, or a corrective feedback, consisting in a set of actions and guards that should be added to the operation in order to make it SE.

3 The Verification Tool

We developed our tool as an Eclipse plugin that can be downloaded from [6].

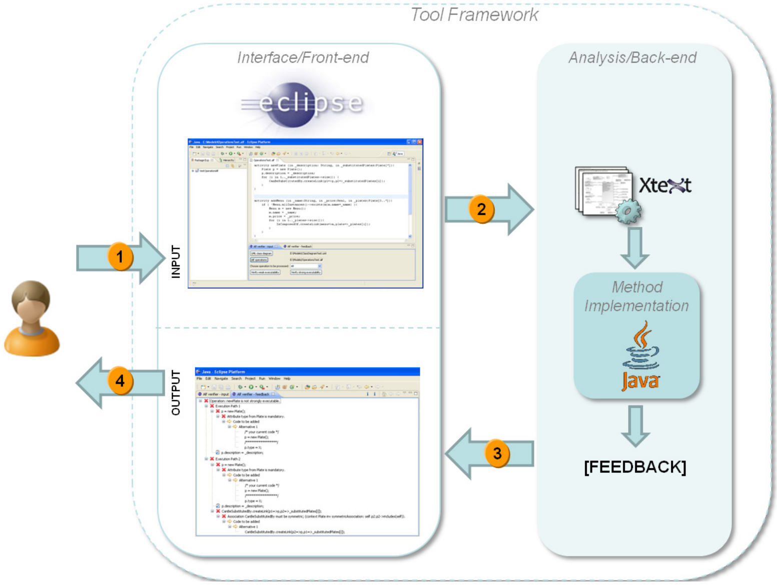
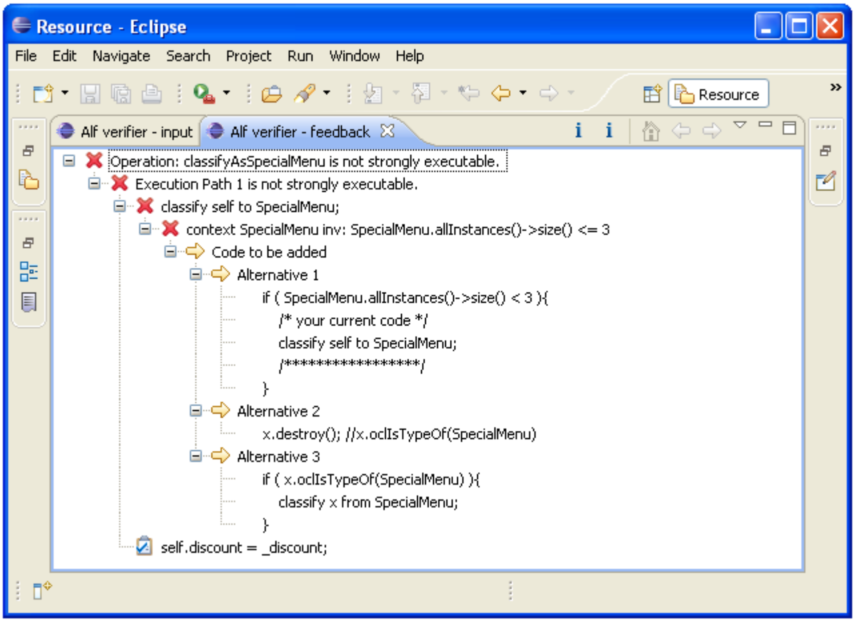


Fig. 2. General architecture or our tool

Figure 2 shows the general view of the tool architecture. As a first step, the designer specifies the UML executable model she wants to deal with. The executable model is composed by: (1) a UML class diagram and a set of OCL integrity constraints modelled using the graphical modelling environment provided by UML2Tools [1], an Eclipse Graphical Modeling Framework for manipulating UML models; and (2) a set of Alf operations specified in a text file with *.alf* extension. Once the executable model is provided, the designer has to click on *"Verify strong executability"* button and the core of our method is invoked. This method is implemented as a set of Java classes extended with the libraries of the UML2Tools (to interact with the input UML model) and Xtext [2] (to parse the Alf operations and instantiate them as Java classes). Finally, the feedback provided by our method is displayed, integrated into the Eclipse interface.

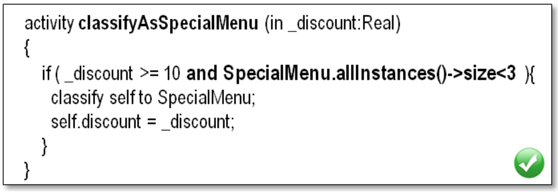
Fig. 3 shows the feedback provided by our tool when verifying the operation `classifyAsSpecialMenu` introduced in Fig. 1. This operation is not strongly executable since the action `classify self to SpecialMenu` may violate the constraint `(2) SpecialMenu.allInstances()->size()`$\leq$`3` (when the system state where the operation is applied already contains three special menus).

To avoid this violation, our tool proposes three alternatives: (1) add a guard ensuring there are less than three special menus; (2) destroy an existing special menu; or (3) reclassify an existing special menu from *SpecialMenu*. Fig. 4 shows the operation once the first alternative (highlighted in bold face) has been applied. After applying this change, we can ensure every execution of the operation `classifyAsSpecialMenu` will be safe.



```
Operation: classifyAsSpecialMenu is not strongly executable.
  Execution Path 1 is not strongly executable.
    classify self to SpecialMenu;
      context SpecialMenu inv: SpecialMenu.allInstances()->size() <= 3
        Code to be added
          Alternative 1
            if ( SpecialMenu.allInstances()->size() < 3 ){
              /* your current code */
              classify self to SpecialMenu;
              /****************/
            }
          Alternative 2
            x.destroy(); //x.oclIsTypeOf(SpecialMenu)
          Alternative 3
            if ( x.oclIsTypeOf(SpecialMenu) ){
              classify x from SpecialMenu;
            }
    self.discount = _discount;
```

Fig. 3. Screenshot of the feedback provided by our Eclipse plugin



```
activity classifyAsSpecialMenu (in _discount:Real)
{
  if ( _discount >= 10 and SpecialMenu.allInstances()->size<3 ){
    classify self to SpecialMenu;
    self.discount = _discount;
  }
}
```

Fig. 4. Operation `classifyAsSpecialMenu` repaired

4 Summary

We have proposed an Eclipse plugin [6] for assisting the designers during the specification of executable behavioural models. Our plugin implements a lightweight method [5] that verifies the *Strong Executability* (SE) of Alf operations wrt the integrity constraints imposed by the class diagram. The main characteristics of our tool are its efficiency (since no simulation/animation of the behaviour is required) and feedback (for non-executable operations, it is able to identify the source of the inconsistency and suggest possible corrections).

Acknowledgements. This work has been partly supported by the Ministerio de Ciencia y Tecnología under TIN2008-00444 project, Grupo Consolidado.

References

1. UML2Tools, http://www.eclipse.org/modeling/mdt/?project=uml2tools (last visit: May 2012)
2. Xtext, www.xtext.org/ (last visit: May 2012)
3. OMG. Concrete Syntax for UML Action Language (Action Language for Foundational UML), version Beta 1 (2010), www.omg.org/spec/ALF
4. OMG. Semantics Of A Foundational Subset For Executable UML Models (fUML), version 1.0 (2011), www.omg.org/spec/FUML
5. Planas, E., Cabot, J., Gómez, C.: Lightweight Verification of Executable Models. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 467–475. Springer, Heidelberg (2011)
6. Planas, E., Sanchez-Mendoza, D.: Alf-verifier: A lightweight tool for verifying UML-Alf executable models (2012), http://code.google.com/a/eclipselabs.org/p/alf-verifier/
7. Stephen, M.J.B., Mellor, J.: Executable UML: A Foundation for Model-Driven Architecture. Addison-Wesley (2002)

A Web-Based Filtering Engine for Understanding Event Specifications in Large Conceptual Schemas

Antonio Villegas, Antoni Olivé, and Maria-Ribera Sancho

Department of Service and Information System Engineering,
Universitat Politècnica de Catalunya – BarcelonaTech,
Barcelona, Spain
{avillegas,olive,ribera}@essi.upc.edu

Abstract. A complete conceptual schema must include all relevant general static and dynamic aspects of an information system. Event types describe a nonempty set of allowed changes in the population of entity or relationship types in the domain of the conceptual schema. The conceptual schemas of many real-world information systems that include the specification of event types are too large to be easily managed or understood. There are many information system development activities in which people need to understand the effect of a set of events. We present an information filtering tool in which a user focuses on one or more event types of interest for her task at hand, and the tool automatically filters the schema in order to obtain a reduced conceptual schema that illustrates all the elements affected by the given events.

Keywords: Large Schemas, Filtering, Event Types, Importance.

1 Introduction

The conceptual schemas of many real-world information systems include the specification of event types. An event describes a nonempty set of changes in the population of entity or relationship types in the domain of the conceptual schema. The sheer size of those schemas makes it difficult to extract knowledge from them. There are many information system development activities in which people need to understand the effect of a set of events. For example, a software tester needs to write tests checking that the effect of an event has been correctly implemented, or a member of the maintenance team needs to change that effect. Currently, there is a lack of computer support to make conceptual schemas usable for the goals of event exploration and event understanding.

The aim of information filtering is to expose users only to information that is relevant to them [1]. We present an interactive tool in which the user specifies one or more event types of interest and the tool automatically provides a (smaller) subset of the knowledge contained in the conceptual schema that includes all the elements affected by the given events. The user may then start another interaction with different events, until she has obtained all knowledge of interest. We presented the theoretical background behind this tool in [2].

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 383–386, 2012.

2 Events as Entities

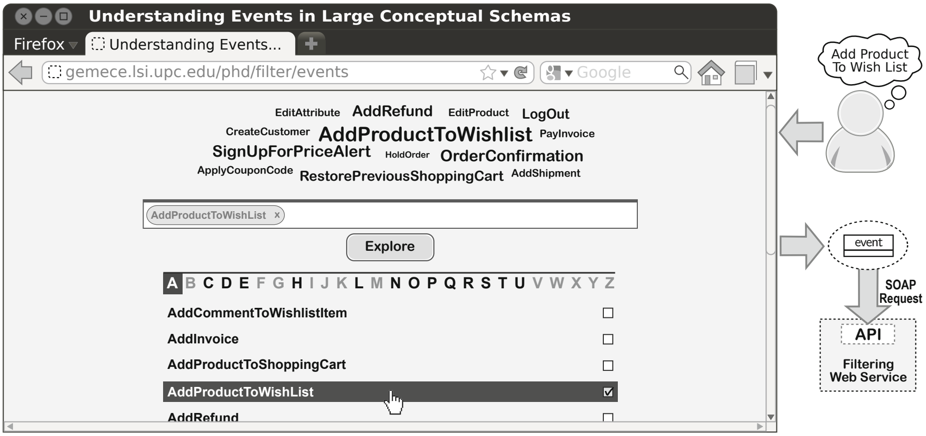
We adopt the view that events are similar to ordinary entities and, therefore, that events can be modeled as a special kind of entities [3]. In the UML, we use for this purpose a new stereotype, that we call «event». A type with this stereotype defines an event type. Like any other entity type, event types may be specialized and/or generalized. This will allow us to build a taxonomy of event types, where common elements are defined only once.

The characteristics of events should be modeled like those of ordinary entities. In the UML, we model them as attributes or associations. We define a particular operation in each event type, whose purpose is to specify the event effect. To this end, we use the operation *effect*. The pre- and postconditions of this operation will be exactly the pre- and postconditions of the corresponding event. We use the OCL to specify these pre- and postconditions formally.

3 The Filtering Engine for Events

In this section we describe how the event specifications of a large conceptual schema can be explored by using our tool, which corresponds to the demonstration we intend to perform. The main idea is to extract a reduced and self-contained view from the large schema, that is, a filtered schema with the elements affected by the specification of a set of events.

Our filtering tool is developed as a web client that interacts with a web service following the SOAP protocol. The filtering web service we have implemented uses a customized version of the core of the USE tool [4] to access the knowledge of the large schema the user wants to explore. In our demonstration, we use the schema of the Magento e-commerce system, which contains 218 entity types, 187 event types, 983 attributes, 165 generalizations, and 319 associations [5].


Understanding Events in Large Conceptual Schemas
Firefox
Understanding Events...
gemece.lsi.upc.edu/phd/filter/events
Google
EditAttribute AddRefund EditProduct LogOut
CreateCustomer AddProductToWishlist PayInvoice
SignUpForPriceAlert HoldOrder OrderConfirmation
ApplyCouponCode RestorePreviousShoppingCart AddShipment
AddProductToWishList ×
Explore
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Add Product To Wish List
event
SOAP Request
API
Filtering Web Service

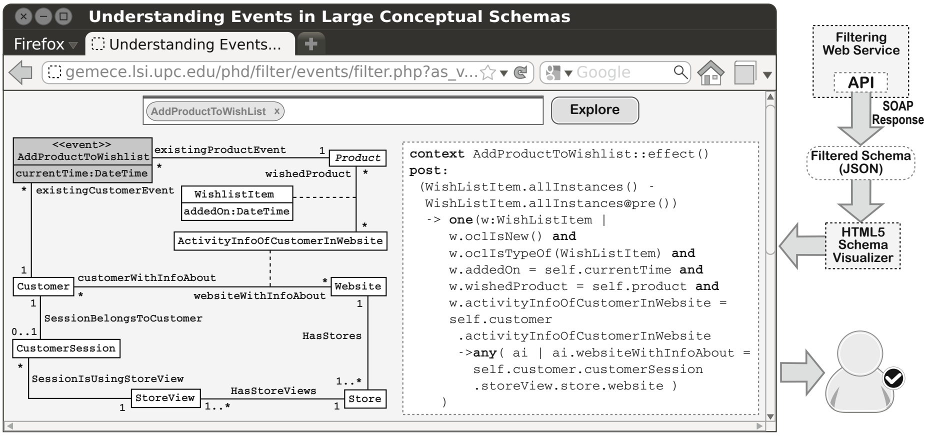
AddCommentToWishlistItem □
AddInvoice □
AddProductToShoppingCart □
AddProductToWishList ☑
AddRefund □

Fig. 1. Request of a user to our filtering tool

Figure 1 presents the main components of the filtering request. First, the user focus on a nonempty set of event types she is interested in according to a specific information need over the event specifications of a large schema. These events conform the input focus set our web-based filtering engine needs to start the process. To help the user on selecting the events of focus, our web client provides a word cloud with the names of the top event types according to their general relevance in the schema, and an alphabetical list with the names of all the event types. The user can select events from both components, or directly write the name of the event in the search bar placed between them. It automatically suggests event names while the user is typing. Once the request is submitted, the web client constructs a SOAP request and sends it to the API of the filtering web service with the focus set of events. As an example, the user focus on the event type *AddProductToWishList* in Fig. 1.

Figure 2 presents the main components of the filtering response. The web service of the filtering engine automatically obtains the corresponding pre- and postconditions for the events of the request. The OCL specification of these constraints references the elements from the large schema that are affected by the events of focus. Our service processes the OCL expressions of the previous constraints in order to extract all the elements that appear within their formal specification. Then, it puts them together with the event types of the focus set in order to create a filtered schema with the elements of both sets. The main goal consists of producing a filtered schema of minimum size. To achieve this, our tool projects the referenced attributes and relationship types used in the event specifications that are defined in the context of elements that are not referenced by the OCL expressions to entity or event types in the filtered schema, whenever possible. It also connects the elements of the filtered schema with generalizations according to the knowledge in the large schema [2].

As a result, the web service returns the corresponding filtered schema in JSON format through a SOAP response in less than one second. Then, we use an



```
context AddProductToWishlist::effect()
post:
 (WishListItem.allInstances() -
  WishListItem.allInstances@pre())
  -> one(w:WishListItem |
     w.oclIsNew() and
     w.oclIsTypeOf(WishListItem) and
     w.addedOn = self.currentTime and
     w.wishedProduct = self.product and
     w.activityInfoOfCustomerInWebsite =
     self.customer
      .activityInfoOfCustomerInWebsite
      ->any( ai | ai.websiteWithInfoAbout =
        self.customer.customerSession
        .storeView.store.website )
    )
```

Fig. 2. Response of our filtering tool to the user

HTML5-based schema visualizer to graphically represent the filtered schema in UML, including the textual representation in OCL of the event effects. Thus, the user is able to interact with the resulting fragment of the large schema directly from the web browser. In the example, the user can easily understand that the effect of the event *AddProductToWishList* creates a new instance of *WishListItem* involving the product and the customer associated with the event, in order to represent the addition of a product into a wish list in Magento. The new instance is related to the activity information of the customer that wants to add the product to her wish list. It includes the customer session in the store view of the store owned by the website of the system. Note that the event is marked in gray in order to rapidly identify the focus of the filtered schema. Subsequently, the user can start a new filtering request if required.

4 Summary

We have presented a tool that automatically obtains a reduced view of the schema, which includes the relevant parts for the understanding of the events. Our implementation as a web service provides interoperability and simplifies the interaction with users. A preliminary version of the filtering tool can be found in `http://gemece.lsi.upc.edu/phd/filter/events`.

Our immediate plans include the improvement of our tool by providing traceability between the graphical representation of the elements within the filtered schema and their references in the OCL expressions that specify the effect of the events of focus. As a result, the user will be able to quickly understand the formal specification of the events and their impact in the schema.

Acknowledgements. Our thanks to Jose Maria Gomez for his work in the implementation of the schema visualizer. This work has been partly supported by the Ministerio de Ciencia y Tecnologia and FEDER under project TIN2008-00444/TIN, Grupo Consolidado, and by Universitat Politècnica de Catalunya under FPI-UPC program.

References

1. Hanani, U., Shapira, B., Shoval, P.: Information filtering: Overview of issues, research and systems. User Modeling and User-Adapted Interaction 11(3), 203–259 (2001)
2. Villegas, A., Olivé, A.: A Method for Filtering Large Conceptual Schemas. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 247–260. Springer, Heidelberg (2010)
3. Olivé, À.: Definition of Events and Their Effects in Object-Oriented Conceptual Modeling Languages. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, vol. 3288, pp. 136–149. Springer, Heidelberg (2004)
4. Gogolla, M., Büttner, F., Richters, M.: USE: A UML-based specification environment for validating UML and OCL. Science of Computer Programming (2007)
5. Ramirez, A.: Conceptual schema of Magento. Technical report, Universitat Politècnica de Catalunya (2011), `http://hdl.handle.net/2099.1/12294`

An Eclipse Plugin for Improving the Quality of UML Conceptual Schemas

David Aguilera, Cristina Gómez, and Antoni Olivé

Department of Service and Information System Engineering,
BarcelonaTech – Universitat Politècnica de Catalunya,
Barcelona, Spain
{daguilera,cristina,olive}@essi.upc.edu

Abstract. The development of an information system requires its conceptual schema to be of high quality. Classically, this quality comprises properties such as syntactic and semantic correctness, relevance, and completeness, but many other quality properties have been proposed in the literature. In this demonstration we integrate some published quality properties in Eclipse by extending the core functionalities of MDT. These properties include syntactic issues, naming guidelines, and best practices. A quality property is defined using OCL and is specified in an XML file. The set of quality properties included in our tool is available on an online public catalog that can be extended to include new quality properties. We use XSLT to present this catalog in a friendly manner to users that access it using a web browser.

Keywords: Eclipse, UML, Quality Properties, Web-based Catalog, Conceptual Schemas.

1 Introduction

The development of an information system requires its conceptual schema to be of high quality [1]. This quality can be measured in terms of syntactic and semantic correctness, relevance and completeness, and many other properties that have been proposed in the literature. For example, conceptual modelers may improve the quality of their conceptual schemas by applying a certain naming guideline [2], or may improve its understandability (and, thus, its quality as well) by following some best practices—such as refactoring the schema [3] or making implicit constraints explicit [4].

Current UML modeling environments offer little or no support to practitioners on checking and improving the quality of a conceptual schema. In general, these environments check syntactic properties only, putting aside most of the work available with regard to quality and understandability. In this demonstration, we present an Eclipse plugin[1] that integrates some of the aforementioned quality properties in the environment. Each property is defined using OCL in an XML file and is included in an online public catalog. As a result, our plugin can load a set of up-to-date quality properties dinamically so that practitioners can benefit

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 387–390, 2012.

from them automatically. We use XSLT to present this catalog in a friendly manner to those users that access it using a web browser.

2 Eclipse Plugin to Support Quality Issues

Eclipse is an extensible platform that consists of, on the one hand, a small kernel that provides core services and, on the other hand, a set of plugins that defines and extends its functionalities [5]. In particular, Eclipse's Model Development Tools (MDT) is a set of plugins that converts Eclipse into a modeling environment. Among the main functionalities provided by MDT, there is a UML model editor (UML2 Tools) and an OCL interpreter.

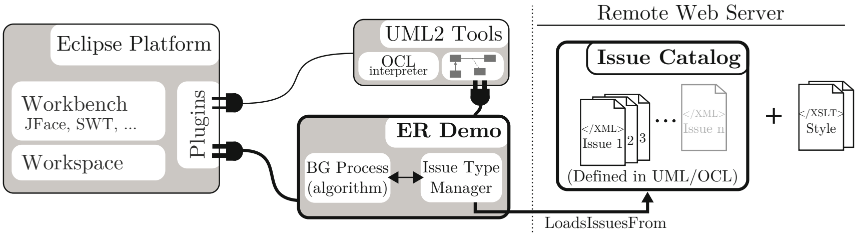


Fig. 1. Architecture of our Eclipse plugin

We conceived our tool as an Eclipse plugin that extends the UML2 Tools framework and uses the MDT's OCL interpreter. Figure 1 depicts its architecture. First, the plugin loads into the Eclipse Platform a set of quality properties defined in OCL (i.e. "issues") from an issue catalog—which, as we shall see in the next section, is defined in a remote, public server. Then, when a modeler is working with a conceptual schema, our tool evaluates the OCL expressions to detect which issues are present in the schema. This process is performed in the background so that the modeler is not disturbed during her modeling tasks.

For example, a naming guideline stating that "Classes must be capitalized" could be formally defined as the following OCL expression

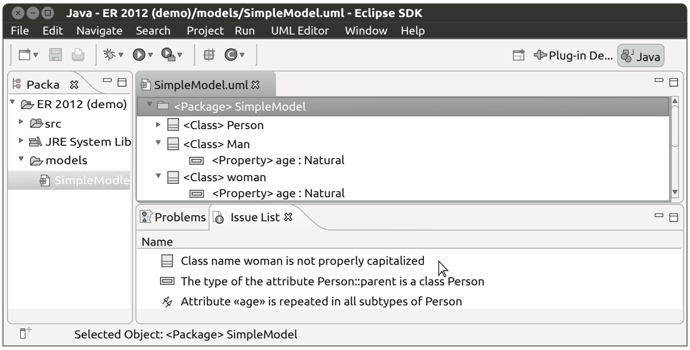
```
context Class:
   self.name.at(1) <> self.name.at(1).toUpperCase()
```

which would be checked for every single instance of *Class*. When the condition is *true* for a certain instance i, our tool detects that the issue exists for i and the modeler is notified.

In order to evaluate an issue's OCL condition, our plugin uses the MDT's OCL interpreter. There are some situations in which defining an issue condition in OCL is too difficult or unfeasible: for example, (a) detecting if a constraint defined in OCL is syntactically correct (i.e. it compiles), or (b) checking whether "a class name is a noun phrase whose head is countable". In these situations, we would like to define additional helper operations that simplify the definition of

[1] The plugin can be downloaded from http://helios.lsi.upc.edu/phd/downloads

OCL expressions. The OCL interpreter included in MDT can be easily extended by implementing additional operations in Java which can be later used in an issue's OCL expression. A couple of examples of these additonal helper operations include (a) *String*::*compiles*():*Boolean*, which returns if a *String* representing an OCL expression is syntactically correct or not, and (b) *String*::*getHead()*:*String*, which returns the head of a noun phrase.


Java - ER 2012 (demo)/models/SimpleModel.uml - Eclipse SDK
File Edit Navigate Search Project Run UML Editor Window Help
Plug-in De...
Java
Packa
ER 2012 (demo)
src
JRE System Lib
models
SimpleModle
SimpleModel.uml
Problems
Issue List
Name

<Package> SimpleModel
<Class> Person
<Class> Man
<Property> age : Natural
<Class> woman
<Property> age : Natural

Class name woman is not properly capitalized
The type of the attribute Person::parent is a class Person
Attribute «age» is repeated in all subtypes of Person

Selected Object: <Package> SimpleModel

Fig. 2. Screenshot of Eclipse running our plugin

Figure 2 shows a screenshot of our tool. The plugin extends the User Interface (UI) of Eclipse by adding a new view: the Issue List. This view can be shown or hidden by the modeler and contains all the issues that are present in the conceptual schema with which modeler is working. The list is updated periodically in a non-disruptive manner. As a result, the modeler can focus on the modeling tasks, and the issues that arise due to the changes performed in the conceptual schema are always either visible or easily accessible.

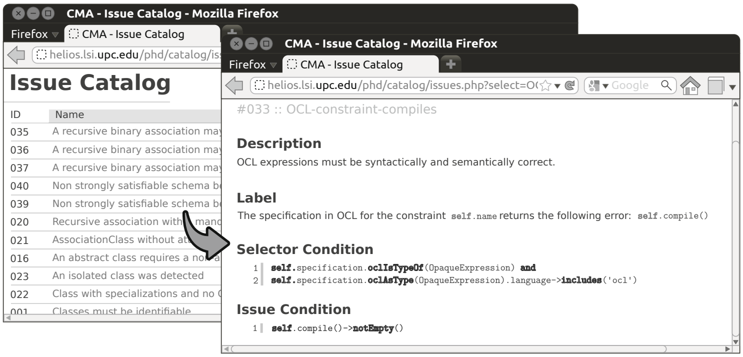
3 Online Catalog of Quality Issues

Our plugin detects the issues that are present in a conceptual schema and provides useful feedback to the modeler. The set of issues the tool is able to deal with is defined in an online public catalog[2] that can be easily extended. Quality issues are defined in XML files that contain information such as the name, the description, or the OCL formal description of each issue. Whenever a new issue is to be added to the system, we only need to provide its XML definition in the catalog.

By externalizing the definition of these issues, conceptual modelers can benefit of a seamless, up-to-date issue catalog in Eclipse, without requiring any updates. Furthermore, third-party tool manufacturers can also take advantadge of the catalog and use it into their environments.

Figure 3 shows two screenshots: one of the issue catalog, and another of the detail of a concrete issue. The issue catalog and issue details are defined as XML files, which benefits both development environments and conceptual modelers. On the

[2] `http://helios.lsi.upc.edu/phd/er2012-catalog/issues.php`


CMA - Issue Catalog - Mozilla Firefox
Firefox
CMA - Issue Catalog
helios.lsi.upc.edu/phd/catalog/is
CMA - Issue Catalog - Mozilla Firefox
Firefox
CMA - Issue Catalog
helios.lsi.upc.edu/phd/catalog/issues.php?select=O(
Google

Issue Catalog

ID	Name
035	A recursive binary association may
036	A recursive binary association may
037	A recursive binary association may
040	Non strongly satisfiable schema be
039	Non strongly satisfiable schema be
020	Recursive association with mand
021	AssociationClass without at
016	An abstract class requires a non a
023	An isolated class was detected
022	Class with specializations and no C
001	Classes must be identifiable

#033 :: OCL-constraint-compiles

Description

OCL expressions must be syntactically and semantically correct.

Label

The specification in OCL for the constraint `self.name` returns the following error: `self.compile()`

Selector Condition

```
self.specification.oclIsTypeOf(OpaqueExpression) and
self.specification.oclAsType(OpaqueExpression).language->includes('ocl')
```

Issue Condition

```
self.compile()->notEmpty()
```

Fig. 3. Screenshot of a web browser displaying our Issue Catalog

one hand, XML-based descriptions permit development environments to easily access, retrieve, parse, and load issues. On the other hand, the Extensible Stylesheet Language Transformations (XSLT) mechanism permits conceptual modelers to browse the catalog in a user-friendly manner when accessed via a web browser.

4 Summary

We have presented a tool to assist modelers during the development of conceptual schemas. The tool is defined as an Eclipse plugin that loads a set of quality properties into the environment and automatically checks the issues that are present in the conceptual schema. The issues are defined in a online public catalog which can be extended to include new issues and, as a result, offers an up-to-date service to modelers and third-party tools.

Acknowledgements. This work has been partly supported by the *Ministerio de Ciencia y Tecnología* and *FEDER* under project TIN2008-00444/TIN, *Grupo Consolidado*, and by *Universitat Politècnica de Catalunya* under FPI-UPC program.

References

1. Maes, A., Poels, G.: Evaluating Quality of Conceptual Models Based on User Perceptions. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 54–67. Springer, Heidelberg (2006)
2. Becker, J., Delfmann, P., Herwig, S., Lis, Ł., Stein, A.: Formalizing Linguistic Conventions for Conceptual Models. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 70–83. Springer, Heidelberg (2009)
3. Fowler, M.: Refactoring: Improving the Design of Existing Code. Addison-Wesley (1999)
4. Costal, D., Gómez, C.: On the Use of Association Redefinition in UML Class Diagrams. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 513–527. Springer, Heidelberg (2006)
5. Clayberg, E., Rubel, D.: Eclipse Plug-ins. Addison-Wesley (2008)

Requirement-Driven Creation and Deployment of Multidimensional and ETL Designs

Petar Jovanovic[1], Oscar Romero[1], Alkis Simitsis[2], and Alberto Abelló[1]

[1] Universitat Politècnica de Catalunya, BarcelonaTech, Barcelona, Spain
{petar,oromero,aabello}@essi.upc.edu
[2] HP Labs, Palo Alto, CA, USA
alkis@hp.com

Abstract. We present our tool, GEM, for assisting designers in the error-prone and time-consuming tasks carried out at the early stages of a data warehousing project. Our tool semi-automatically produces multidimensional (MD) and ETL conceptual designs from a given set of business requirements (like SLAs) and data source descriptions. Subsequently, our tool translates both the MD and ETL conceptual designs produced into physical designs, so they can be further deployed on a DBMS and an ETL engine. In this paper, we describe the system architecture and present our demonstration proposal by means of an example.

1 Introduction

At the early phases of a data warehouse (DW) project, we create conceptual designs for the multidimensional (MD) schema of the DW and the extract-transform-load (ETL) process that would populate this MD schema from the data sources. These labor-intensive tasks are typically performed manually and are known to consume 60% of the time of the overall DW project [12]. Automating these tasks would speed up the designer's work both at the early stages of the project and also, later on, when evolution events may change the DW ecosystem.

Several works have dealt with MD schema modeling and they either focus on incorporating business requirements (e.g., [6,7]) or on overcoming the heterogeneity of the data sources (e.g., [8,11]). Furthermore, it has been noticed that while trying to automate this process, people tend to overlook business requirements or introduce strong constraints (e.g., focus only on relational sources [7]) that typicaly cannot be assumed. On the other hand, several approaches have dealt with ETL design using various techniques like MDA and QVT (e.g., [5]), semantic web technologies (e.g. [10]), and schema mapping (e.g., Clio [4] and Orchid [2]). However, these works do not address the problem of automating the inclusion of the business requirements into the ETL design. To the best of our knowledge, our work is the first toward the synchronous, semi-automatic generation of MD and ETL designs.

Our tool, called GEM, incorporates the business requirements into the design, all the way from the conceptual to the physical levels. The fundamentals behind GEM are described elsewhere [9]. Here, we focus on our system internals and discuss GEM's functionality through an example. In the demonstration, we will show GEM through a number of pre-configured use cases and the conference attendees would be able to interact either by changing the input requirements or by creating designs from scratch.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 391–395, 2012.

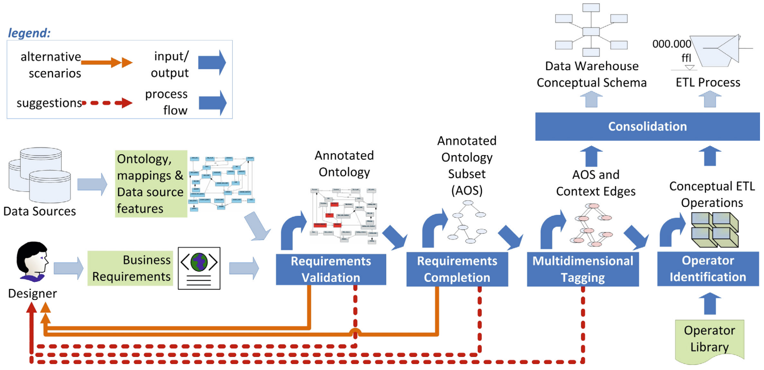


Fig. 1. System overview

2 GEM in a Nutshell

GEM uses an ontology to boost the automation of the design process and produces a conceptual MD design fulfilling the given set of business requirements. At the same time, unlike previous approaches, GEM benefits from the knowledge inferred when producing the MD schema and along with information about the data sources, it automates the production of conceptual ETL design. A high level view of how GEM operates is shown in Figure 1.

Inputs. GEM starts with the *data sources* and *requirements* representing business needs; e.g., service-level agreements (SLAs). First, it maps the data sources onto a domain OWL *ontology* that captures common business domain vocabulary and uses XML to encode the source mappings. It has been shown in [10] that a variety of structured and unstructured data sources can be elegantly represented with an ontology. In addition, the requirements expressing some business needs (e.g., “Revenue for each nation of North Europe region”) are formalized by means of an extensible and generic XML structure (see Figure 3).

Stages. GEM maps each requirement to ontology concepts and further, through the source mappings, to the corresponding data sources (*requirements validation*).Then, by exploring the ontology topology, it identifies the ontology subset needed to retrieve the data concerning the requirement in hand (*requirements completion*). Next, the system produces the complete MD interpretation of the ontology subset (i.e., concepts are either *dimensional* or *factual*), validates the subset respecting MD paradigm and generates the conceptual design of the output MD schema (*multidimensional tagging and validation*). Finally, by considering the MD schema knowledge inferred during the previous stage and how the concepts are mapped to the sources, it identifies the ETL operations needed to populate the MD schema from the sources (*operation identification*).

Physical designs. After having produced the MD and ETL designs, we translate each design to a physical model. Due to space limitation, we omit the technical details (see [13]), but in the demonstration we will show how GEM connects to a DBMS for creating and accessing the MD schema and to an ETL engine for creating an ETL flow.

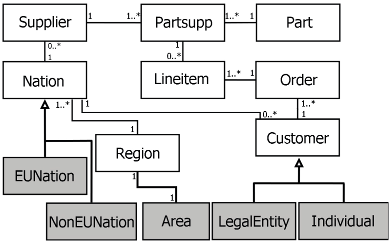
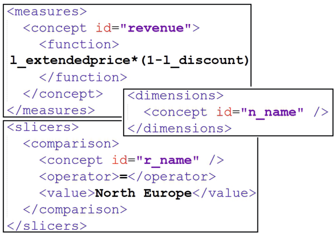


Fig. 2. Example ontology for the TPC-H schema



```
<measures>
  <concept id="revenue">
    <function>
l_extendedprice*(1-l_discount)
    </function>
  </concept>
</measures>
<dimensions>
  <concept id="n_name" />
</dimensions>
<slicers>
  <comparison>
    <concept id="r_name" />
    <operator>=</operator>
    <value>North Europe</value>
  </comparison>
</slicers>
```

Fig. 3. Example requirement in XML

Implementation. GEM is implemented in Java 1.6. We use JAXP - SAX API for parsing XML files and JENA for parsing OWL ontologies. The interface is implemented using Java Swing library. In its current implementation, GEM connects to a DBMS (like PostgreSQL) for storing and accessing database constructs and uses Pentaho Data Integration (PDI) as an ETL execution engine.

3 Demonstration Scenario

Our on-site demonstration will involve several use cases. Each case is pre-configured so that would help us demonstrate individual characteristics (e.g., variety and complexity of MD and ETL designs, variety of business requirements, and so on). However, here due to space considerations, we limit ourselves into a single use case that represents two problems typically encountered in real-world DW projects: (P1) the information at hand for data sources is incomplete and (P2) the business requirements are ambiguous.

Our example is based on the *TPC-H benchmark* [1]. First, the domain ontology (Figure 2), describing the TPC-H sources is enriched with the business domain vocabulary (shown as shaded elements in Figure 2). Then, we consider the mappings of the ontology concepts to the data sources in an iterative fashion. For (P1), we consider a mapping where the concept *nation* is not mapped to any source. In addition, we create the input XML representing business requirements. For (P2), we consider an ambiguous requirement as shown in the snippet depicted in Figure 3: *“Revenue for each nation of North Europe region”*.

During the requirements validation stage, GEM identifies requirement concepts (i.e., *lineitem*, *nation*, and *region*) as MD concepts and checks how they map to the sources. Since, the concept *nation* is not mapped to any data source the system tries to map it by looking for synonyms (1-1 relationships) and exploring concept’s taxonomies inside the ontology. In this case, GEM suggests mapping *nation* through its mapped subclasses (i.e., *EUNation* and *NonEUNation*).

In the requirements completion stage, GEM identifies that due to ambiguous business requirements the concepts *nation* and *lineitem* may be related either through the concept *customer* or through *supplier*; i.e., the revenue of *customers* or the revenue of *suppliers* may be of interest to the business user. The designer is informed about this ambiguity and is asked to identify the appropriate semantics. After a path is chosen (e.g., through *supplier*), GEM produces the suitable ontology subset.

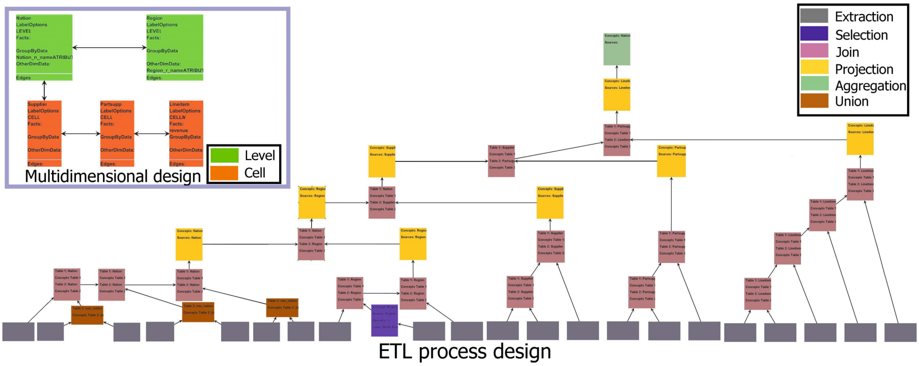


Fig. 4. GEM output designs

Next, GEM checks for a sound MD interpretation of the produced subset and eventually, produces a UML-like conceptual MD schema design fulfilling the input requirement (as shown in the top left part of Figure 4). Finally, for each mapped concept, GEM produces an *extraction* operation and, in case of derived mappings, such as *nation*, the proper operator (e.g., *union*) over the corresponding *extraction* operators (e.g., *EUNation* and *NonEUNation*) is added. Similarly, the slicer on *region* is translated as a *selection* operation and the remaining ETL operators (e.g., *joins*, *projections*, and *aggregations*) needed to produce the data cube described by the ontology subset are also added. Figure 4 shows the ETL design for this case.

The interested reader may see a detailed walkthrough of this use case with snapshots of the tool in a web page we have set up (see [3]). In the web page, we also show the corresponding physical designs for both MD and ETL designs.

References

1. TPC-H, `http://www.tpc.org/tpch/spec/tpch2.14.0.pdf`
2. Dessloch, S., Hernández, M.A., Wisnesky, R., Radwan, A., Zhou, J.: Orchid: Integrating schema mapping and etl. In: ICDE, pp. 1307–1316 (2008)
3. GEM snapshots, `http://www.essi.upc.edu/~petar/demo.html`
4. Haas, L.M., Hernández, M.A., Ho, H., Popa, L., Roth, M.: Clio grows up: from research prototype to industrial tool. In: SIGMOD Conference, pp. 805–810 (2005)
5. Muñoz, L., Mazón, J.N., Trujillo, J.: Automatic generation of etl processes from conceptual models. In: DOLAP, pp. 33–40 (2009)
6. Nabli, A., Feki, J., Gargouri, F.: Automatic construction of multidimensional schema from olap requirements. In: AICCSA, p. 28 (2005)
7. Romero, O., Abelló, A.: Automatic Validation of Requirements to Support Multidimensional Design. Data Knowl. Eng. 69(9), 917–942 (2010)
8. Romero, O., Abelló, A.: A framework for multidimensional design of data warehouses from ontologies. Data Knowl. Eng. 69(11), 1138–1157 (2010)
9. Romero, O., Simitsis, A., Abelló, A.: *GEM*: Requirement-Driven Generation of ETL and Multidimensional Conceptual Designs. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 80–95. Springer, Heidelberg (2011)
10. Skoutas, D., Simitsis, A.: Ontology-based conceptual design of etl processes for both structured and semi-structured data. Int. J. Semantic Web Inf. Syst. 3(4), 1–24 (2007)

11. Song, I., Khare, R., Dai, B.: SAMSTAR: A Semi-Automated Lexical Method for Generating STAR Schemas from an ER Diagram. In: DOLAP, pp. 9–16 (2007)
12. Vassiliadis, P., Simitsis, A.: Extraction, transformation, and loading. In: Encyclopedia of Database Systems, pp. 1095–1101 (2009)
13. Wilkinson, K., Simitsis, A., Castellanos, M., Dayal, U.: Leveraging Business Process Models for ETL Design. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 15–30. Springer, Heidelberg (2010)

STS-Tool: Using Commitments to Specify Socio-Technical Security Requirements

Elda Paja, Fabiano Dalpiaz, Mauro Poggianella, Pierluigi Roberti, and Paolo Giorgini

Department of Information Engineering and Computer Science,
University of Trento, Italy
{paja,dalpiaz,poggianella,roberti,giorgini}@disi.unitn.it

Abstract. In this paper, we present STS-Tool, the modelling and analysis support tool for STS-ml, an actor- and goal-oriented security requirements modelling language for Socio-Technical Systems (STSs). STS-Tool allows designers to model a socio-technical system at a high-level of abstraction, while expressing constraints (security needs) over the interactions between the actors in the STS, and derive security requirements in terms of social commitments (promises with contractual validity) once the modelling is done.

1 Introduction

Socio-Technical Systems (STSs) are an interplay of social (human and organisations) and technical subsystems, which interact with one another to reach their objectives, making a STS a network of social relationships. Each subsystem is a participant of the STS, and interacts with others through *message exchange*. But, participants in STSs are autonomous, heterogeneous and weakly controllable. This raises up a number of security issues when they interact, especially when interaction involves information exchange, since one might want to constrain the way information is to be manipulated by others. To deal with social aspects of the security problem in STSs, we have proposed to use *social commitments* [4] to constrain interaction. *Social commitments* are a purely social abstraction used to model interaction. They exist as a result of interaction: they are created and evolve while agents exchange messages.

The focus of our work is on security requirements engineering (SRE) for STSs, while allowing interacting parties to express their *needs* regarding security. We have proposed STS-ml [1] (Socio-Technical Security modelling language), an actor- and goal-oriented security requirements modelling language for STSs, to use our idea of relating security requirements to interaction. The language allows actors to express *security needs* over interactions to constrain the way interaction is to take place, and uses the concept of *social commitment* among actors to specify security requirements.

The notion of *social commitments* was first proposed by Singh [4], and we specialise it for the first time to represent security requirements. Other approaches to SRE either rely on high-level abstractions, such as goals or softgoals [3], or on technical mechanisms such as monitoring [2]. Instead, we concentrate on securing the interaction between actors. An important feature of *social commitments* that makes them adequate for this purpose, is that they have *contractual validity*. That is, non satisfaction of a

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 396–399, 2012.

social commitment might lead to further commitments to be made by the violator. In STS-ml they are used as a guarantee for the satisfaction of *security needs*: a commitment is made by an actor (*responsible*) to another actor (*requestor*) for the satisfaction of a *security need*. For instance, in e-commerce, a buyer (*requestor*) might want a seller not to disclose its credit card details to other parties, and to use this information strictly to perform the payment of the acquired goods. Once the buyer expresses these needs, the seller (*responsible*) commits to him that his credit card details will not be disclosed to other parties, and will be used only for the payment of the acquired goods. The list of *social commitments* is derived for each *security need* expressed by the stakeholders, and represents the security requirements specification for the system-at-hand. They prescribe the security properties stakeholders have to comply with in order for their interactions (and the STS) to be secure.

In this paper, we illustrate the usage of the concept of *social commitment* for the specification of security requirements. Specifically, we show how STS-Tool[1], the graphical modelling and analysis support tool for STS-ml, enables the derivation of security requirements expressed as *social commitments*.

2 Demonstration Content

Our demonstration will cover three main activities. *First*, we will show STS-Tool, the tool that supports modelling activities and the derivation of security requirements as proposed in STS-ml. STS-ml supports multi-view modelling: interactions among actors can be represented by focusing on orthogonal views. As shown in Fig. 1, STS-ml consists of three different views: *social*, *authorisation*, and *information*. The *security needs* are expressed in the *operational view* (Fig. 1), which consists of the three aforementioned views. The *operational view* is automatically mapped to the specification of *security requirements*, which supports the *security needs* expressed in the *operational view*. STS-Tool supports this feature, by providing different views on a diagram, showing specific elements while hiding others depending on the view one is working on. It performs *consistency checking* to help designers create diagrams that follow the semantics of STS-ml. Once the modelling is done, the tool offers designers the possibility to export the diagram (or the different views) to different file formats, such as png, etc.

Second, we will show the use of *social commitments* in serving as specification of security requirements for the system-to-be. For this purpose, we will show small examples to better explain how we capture interactions in STS-ml and how we derive the specification of security requirements.

Finally, we will show an already modelled scenario from a case study on e-Gov, developed as part of the European research project Aniketos [2]. The focus of this part of the demo will be on two aspects: *derivation of security requirements* and *generation of security requirements document*. For a more interactive demo, we will illustrate the features of STS-Tool by modelling a small scenario from the case study (*Example 1*).

[1] STS-Tool is available for download at `http://fmsweng.disi.unitn.it/sts`

[2] `http://www.aniketos.eu/`

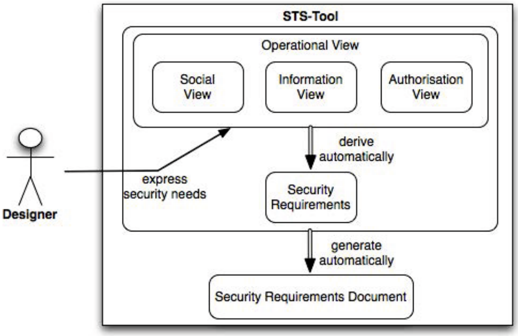


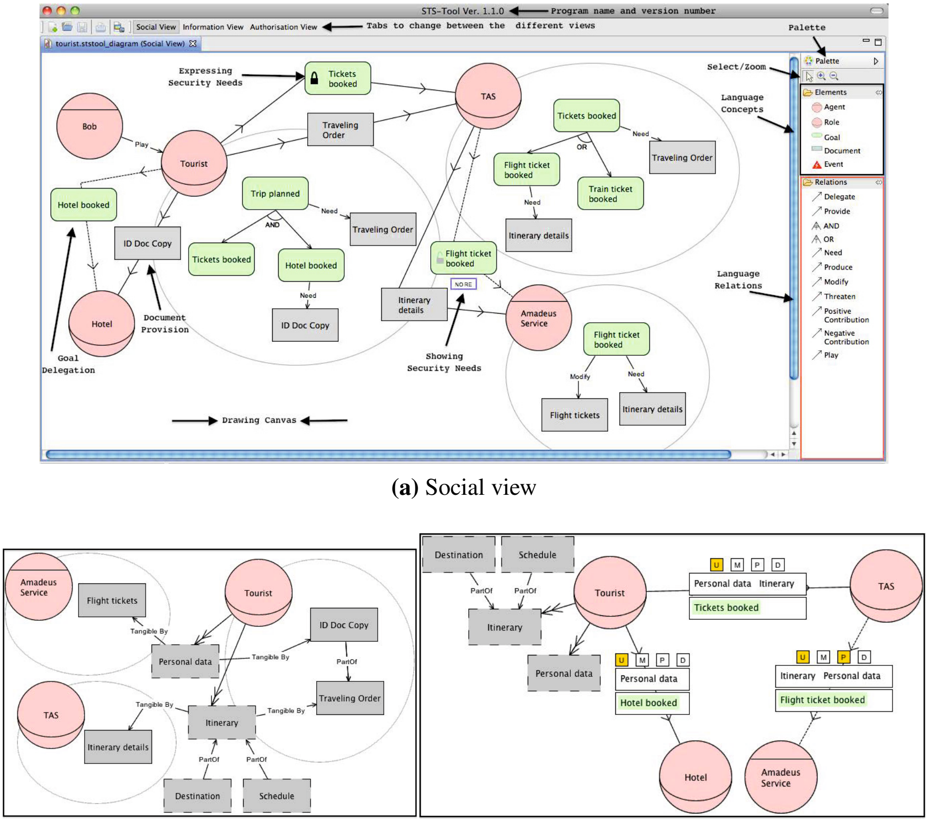
Fig. 1. From the operational view to security requirements

Example 1. Land selling involves not only finding a trustworthy buyer, but also exchanging several documents with various governmental bodies. The seller needs the municipality to certify that the land is residential zoning. The land selling process we consider is supported by an eGov application, through which the official contract (including the municipalitys certification) is sent to the ministry (who has the right to object) and is archived.

We will follow an iterative modelling process to model the different views: *social*, *information*, and *authorisation* view (Fig. 2) for the illustrating scenario. This will help us show how the tool facilitates and supports the modelling process.

Derivation of security requirements: we will show how the list of security requirements for the modelled scenario is derived once the modelling is done. STS-Tool allows the automatic derivation of security requirements, which are provided in a tabular form. The security requirements are listed, and they make clear the difference between actors that *request* a certain security need from those that are *responsible* for satisfying it. Security requirements can be sorted or filtered according to their different attributes. For instance, filtering the security requirements with respect to the *responsible* actor, gives an idea of who are the actors responsible to satisfy the commitments. On the other hand, filtering security requirements according to the *requirement type*, groups together commitments that need to be satisfied to fulfil a certain security need.

Generation of security requirements document: at the end of the modelling process, the tool allows designers to export models and generate automatically a *security requirements document*, which helps them communicate with stakeholders (Fig. 1). This document is customisable: designers can choose among a number of model features to include in the report (e.g., including only a subset of the actors, concepts or relations he wants more information about). However, the overall document provides a description of STS-Tool and communicates security requirements by providing details of each STS-ml view, together with their elements. The diagrams are explained in detail providing textual and tabular description of the models. The document is organised in sections, which the designer can decide to include or not in the generated document.



(a) Social view

(b) Information view

(c) Authorisation view

Fig. 2. Multi-view modelling for the eGov scenario

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant no 257930 (Aniketos) and 256980 (NESSoS).

References

1. Dalpiaz, F., Paja, E., Giorgini, P.: Security requirements engineering via commitments. In: Proceedings of the First Workshop on Socio-Technical Aspects in Security and Trust (STAST 2011), pp. 1–8 (2011)
2. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modeling security requirements through ownership, permission and delegation. In: Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE 2005), pp. 167–176 (2005)
3. Liu, L., Yu, E., Mylopoulos, J.: Security and Privacy Requirements Analysis within a Social Setting. In: Proceedings of the 11th IEEE International Conference on Requirements Engineering (RE 2003), pp. 151–161. IEEE Computer Society (2003)
4. Singh, M.P.: An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts. Artificial Intelligence and Law 7, 97–113 (1999)

CAWE DW Documenter: A Model-Driven Tool for Customizable ETL Documentation Generation[*,**]

Robert Krawatzeck, Frieder Jacobi, and Marcus Hofmann

Chemnitz University of Technology, Chemnitz, Germany
{robert.krawatzeck,frieder.jacobi,marcus.hofmann}
@wirtschaft.tu-chemnitz.de

Abstract. Within business intelligence systems (BI systems), ETL (extract, transform and load) processes move numerous data from heterogeneous sources to a data warehouse and become more complex with growing enterprise size. To keep costs and expenditure of time for maintenance and evolution of those systems slight, ETL processes should be documented. A well-documented system also leads to higher transparency regarding the origin and processing of data, which increases the system's acceptance by business users. However, the preparation of high quality software documentation is sophisticated and therefore it usually only takes place in the design or development phase of BI systems. To ensure that the documentation is always updated, automated generation is advantageous. The paper at hand presents the research prototype CAWE DW Documenter for automated configurable ETL documentation generation.

Keywords: ETL processes, user-specific documentation, Computer-Aided Data Warehouse Engineering, MDA, ADM, reverse engineering, software prototype.

1 Introduction

In the field of business intelligence (BI), extract, transform, load (ETL) refers to three separate functions combined in a single software component. First, the extract function reads data from a specified source system and extracts a desired subset of data. Next, the transform function works with the acquired data converting them into the desired state. Finally, the load function is used to write the resulting data to a tar-get system (e.g. data warehouse, DW), which may or may not previously exist [1].

With an increasing enterprise size, the number of source and target system rises, leading to more complex ETL processes. At the same time, end users want to be informed about how facts are determined, so they feel certain that they base their decision on correct information. With growing complexity of ETL processes, providing

[*] This research has been partly funded by the European Social Fund and the Federal State of Saxony, Germany.

[**] A proof-of-concept video – which demonstrates a former research state using the example of the open source ETL modeling tool Pentaho Data Integration – is available at
`http://www.cawe.eu/demo/20120127-POC-CAWE-DW-Documenter.wmv`

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 400–403, 2012.

this information becomes more difficult. This is enhanced by the fact that most end users do only need a part of the data embedded in ETL process descriptions. To meet the needs of different user roles and organizational units, the automated process for creating documentation has to be configurable.

The requirements on BI systems are predetermined by a complex corporate environment and therefore underlie changes over time. This leads to a permanent process of adaption. As a result, the documentation of ETL processes which has been defined in the design and concept phase respectively in the development process become obsolete over time and are finally no longer valuable [2].

An empirical study conducted in 2011 [3, 4] shows that automated documentation of ETL processes is not properly supported by methods and tools, although ETL processes have a high share in the development expenses of BI systems [5].Therefore, the paper at hand presents the CAWE DW Documenter, a research prototype that satisfies the need for a vendor-independent support for an automated ETL process documentation. It therefore closes the scientific gap in the field of customizable automatic ETL documentation generation.

2 Underlying Technologies

The CAWE DW Documenter combines two different model-driven approaches: (1) Computer-Aided Data Warehouse Engineering (CAWE) – an approach for BI systems reengineering which itself is based on the Model Driven Architecture (MDA) and the Architecture Driven Modernization (ADM) – and (2) a generic model-driven framework for automated generation of user-specific IT systems documentation.

CAWE [6] is an approach for the development of maintainable BI systems and the evolution, migration and reengineering of existing ones.

To assure high quality documentation as the resulting artifact of re-documentation, the CAWE DW Documenter adapts a generic framework for automated generation of user-specific IT systems documentation [7] to the ETL domain [8]. For details and further references of the framework which is now implemented within Eclipse Modeling Framework (EMF) see [7, 8].

3 The CAWE DW Documenter

The CAWE DW Documenter is able to generate documentation for ETL processes modeled in various tools. Using a proprietary, generic and platform-independent ETL meta-model, documentation generation is practically vendor-independent and allows further metadata analyses. Using the prototype is quite simple. After launching the Eclipse IDE, we'll use the *"New Documentation Project"* wizard which leads through the process of gathering the relevant parameters. The user first specifies the location of a certain tool-specific ETL process description. Currently supported input formats include Microsoft SQL Server Integration Services (SSIS) files, Pentaho Data Integration (PDI) files and PDI ETL process descriptions stored in a database. The second parameter defines the desired documentation output format. Although the CAWE DW

Documenter is designed to support various output formats (e.g. Wiki, PDF, HTML), the focus of the following paper will be placed on the generation of MediaWiki files. Therefore, the Wiki's URL and login credentials must be passed. The wizard now creates a fully configured project including a launch configuration, with which it is possible to generate the fully-featured documentation with one click. Additional settings for user-specific documentation generation may be applied before.

As a result of the generation process, the user receives different MediaWiki pages (e.g. overview pages about the process as a whole, about every transformation step and every data field and variable touched within the transformation) and therefore gains a more detailed insight into the ETL process as provided by ETL modeling tools.

As an example, figure 1 shows the most detailed view on the ETL data flow provided by SSIS. Information on specific fields is only available on a per-step basis. Information about specific variables is not provided. Figure 2 shows in contrast the related documentation generated with the DW Documenter, that provides detailed insight into field definition, lineage and impact (left) as well as variable use (right).

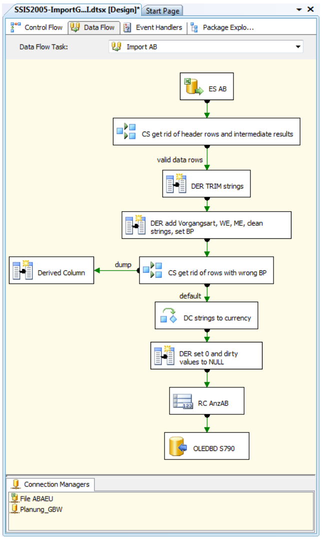
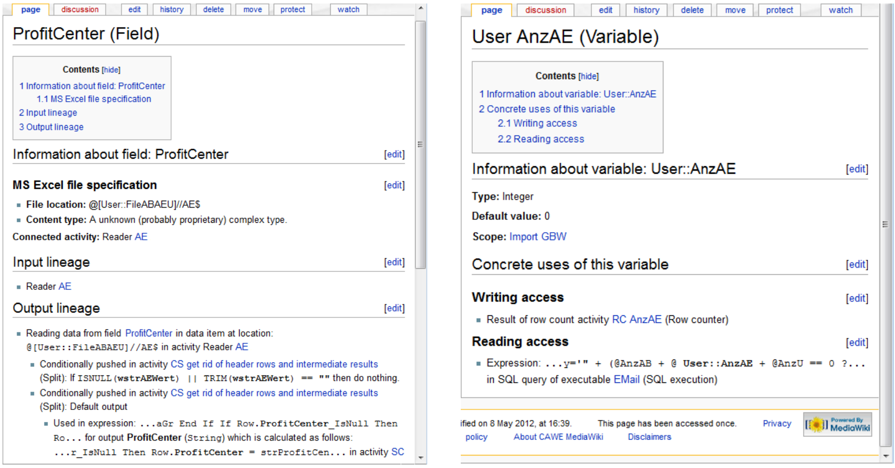


Fig. 1. SSIS ETL data flow



ProfitCenter (Field)

Contents [hide]
1 Information about field: ProfitCenter
1.1 MS Excel file specification
2 Input lineage
3 Output lineage

Information about field: ProfitCenter [edit]

MS Excel file specification [edit]

- **File location:** @[User::FileABAEU]//AE$
- **Content type:** A unknown (probably proprietary) complex type.

Connected activity: Reader AE

Input lineage [edit]

- Reader AE

Output lineage [edit]

- Reading data from field ProfitCenter in data item at location: `@[User::FileABAEU]//AE$` in activity Reader AE
    - Conditionally pushed in activity CS get rid of header rows and intermediate results (Split): If `ISNULL(wstrAEWert) || TRIM(wstrAEWert) == ""` then do nothing.
    - Conditionally pushed in activity CS get rid of header rows and intermediate results (Split): Default output
        - Used in expression: `...aGr End If If Row.ProfitCenter_IsNull Then Ro...` for output **ProfitCenter** (`String`) which is calculated as follows: `...r_IsNull Then Row.ProfitCenter = strProfitCen...` in activity SC

User AnzAE (Variable)

Contents [hide]
1 Information about variable: User::AnzAE
2 Concrete uses of this variable
2.1 Writing access
2.2 Reading access

Information about variable: User::AnzAE [edit]

Type: Integer

Default value: 0

Scope: Import GBW

Concrete uses of this variable [edit]

Writing access [edit]

- Result of row count activity RC AnzAE (Row counter)

Reading access [edit]

- Expression: `...y='" + (@AnzAB + @ User::AnzAE + @AnzU == 0 ?...` in SQL query of executable EMail (SQL execution)

ified on 8 May 2012, at 16:39. This page has been accessed once. Privacy policy About CAWE MediaWiki Disclaimers

Fig. 2. Example MediaWiki pages extracts generated with the CAWE DW Documenter

4 Demonstration Highlights

Within our tool demonstration, we will illustrate the advantages of the CAWE DW Documenter in contrast to built-in documentation capabilities of some ETL tools like

IBM Data Stage or Talend Open Studio. Based on the used model-driven approach and by using additional external information made available by additional metamodels covering IT infrastructure and multidimensional data structures [8], the following values can be added to the generated documentation:

- configurable amount of provided information, based on the target group which the documentation is intended for (effects completeness and definiteness),
- more detailed view on lineage and impact of data fields as well as information about variables by making implicit information explicit by applying further metadata analysis (effects comprehensibility) and
- uniformity through documentation artifacts of different ETL tools (will be demonstrated using the example of the ETL modeling tools SSIS and PDI).

5 Conclusion and Future Work

Currently, the CAWE DW Documenter is a typical proof-of-concept prototype and therefore in an early development stage. As a result, the main functionality is implemented, but the set of supported ETL tools and documentation output formats is small. At the moment of paper creation, we transform the research prototype – with the support of practice cooperation partners like Siemens AG – in a practice-prototype. To ensure the development of a high quality tool supporting maintainable systems, feedback is welcome.

References

1. Vercellis, C.: Business Intelligence: data mining and optimization for decision making. Wiley, Hoboken (2009)
2. Forward, A., Lethbridge, T.C.: The relevance of software documentation, tools and technologies: a survey. In: Proceedings of the 2002 ACM Symposium on Document Engineering (DocEng 2002), pp. 26–33. ACM Press, New York (2002)
3. Gluchowski, P., Hofmann, M., Frieder, J., Krawatzeck, R., Müller, A.: Business-Intelligence-Umfrage 2011: Softwaregestütztes Lebenszyklusmanagement und aktuelles Dokumentationsgeschehen für Business-Intelligence-Systeme (2011), http://www.qucosa.de/recherche/frontdoor/?tx_slubopus4frontend[id]=urn:nbn:de:bsz:ch1-qucosa-75452
4. Hofmann, M., Müller, A., Jacobi, F., Krawatzeck, R.: Umfrage 2011: 'Dokumentation von Business-Intelligence-Systemen' - Ergebnisse und Auswertung. In: Tagungsband der Multikonferenz Wirtschaftsinformatik 2012 (MKWI 2012), Braunschweig, pp. 1091–1104 (2012)
5. Inmon, B.: The data warehouse budget. DM Review Magazine (1997)
6. Kurze, C.: Computer-Aided Warehouse Engineering: Anwendung modellgetriebener Entwicklungsparadigmen auf Data-Warehouse-Systeme. Verlag Dr. Kovač (2011)
7. Krawatzeck, R., Jacobi, F., Müller, A., Hofmann, M.: Konzeption eines Frameworks zur automatisierten Erstellung nutzerspezifischer IT-Systemdokumentationen. In: Workshop Business Intelligence 2011 (WSBI 2011) der GI-Fachgruppe BI, pp. 15–26 (2011)
8. Jacobi, F., Krawatzeck, R., Hofmann, M.: Meeting the Need for ETL Documentation: A Model-driven Framework for Customizable Documentation Generation. In: Proc. of the 18th Americas Conference on Information Systems (AMCIS 2012), Seattle, USA (in print, 2012)

Author Index