

Leipziger Beiträge zur Informatik: Band XII

Subject-centric Computing

Fourth International Conference
on Topic Maps Research and Applications, TMRA 2008
Leipzig, Germany, October 16-17, 2008
Revised Selected Papers

Lutz Maicher
Lars Marius Garshol (Eds.)

Subject-centric Computing

Fourth International Conference
on Topic Maps Research and Applications, TMRA 2008
Leipzig, Germany, October 16-17, 2008
Revised Selected Papers

Volume Editors

Dr. Lutz Maicher

University of Leipzig
Institut für Informatik
Johannisgasse 26, 04103 Leipzig, Germany
E-mail: maicher@informatik.uni-leipzig.de

Lars Marius Garshol

Bouvet AS
Sandakerveien 24C D11
NO-0403 Oslo, Norway
E-mail: larsga@bouvet.no

More information about the TMRA conference and the online versions of all papers within this volume are available at the website: <http://www.tmra.de/2008/>

Subject-centric Computing

Fourth International Conference on Topic Maps Research and Applications,
TMRA 2008 Leipzig, Germany, October 16-17, 2008
Revised Selected Papers
Lutz Maicher, Lars Marius Garshol (Eds.). - Leipzig, 2008

ISBN 978-3-941152-05-2

Preface

The papers in this volume were presented at TMRA 2008, the International Conference on Topic Maps Research and Applications, held 16-17 October 2008, in Leipzig, Germany. TMRA 2008 is the fourth conference in an annual series of international conferences dedicated to Topic Maps in science and industry.

As Peter Brown emphasised in his closing keynote in 2007, traditional information organization is focused on documents, folders, and files, which are all 18th century terms. Topic Maps, on the other hand, are subject-centric, in the sense that they organize information by what it is about. Users, however, typically think and act in a subject-centric way. They don't look for a certain document or folder, but for information about a particular subject that they are interested in. Shifting the information architecture to a subject-centric perspective, means changing the way software and interfaces are designed. Subject-centric computing is based on the appropriate handling of subject identity, which is the mechanism for deciding whether or not two different objects represent the same subject. Identity-aware subject-centric computing empowers a new level of interactivity between systems at global scale. The goal of TMRA 2008 is nothing less than the break-through of subject-centric computing.

Stimulated by the success of the previous conferences the concept of TMRA was retained nearly unchanged. The conference is preceded by tutorials@TMRA 2008, a full day of in-depth tutorials. The main conference schedule is separated into two parallel tracks, providing a rich program for all interests. The Open Space sessions, once more smoothly moderated by Lars Marius Garshol, provide a light and exciting look at work which will most likely be presented at future conferences.

The “Topic Maps Coder Challenge” is a new feature of this year's TMRA conference. NetworkedPlanet and TMRA invited Topic Maps developers to prove their mettle by implementing the Topic Maps and ATOM based protocol for the syndication of semantic descriptions. See the paper of Graham and Küster for more details about the protocol.

An unacceptable delay in the publication of the TMRA 2007 proceedings, combined with pricing and copyright models which are not appropriate for the diffusion of the Topic Maps work presented at the TMRA conference series we have decided to not continue to publish the proceedings in the Springer LNAI series. Instead, the TMRA 2008 proceedings at hand are published in the LIV series of the University of Leipzig. This volume has an ISBN, will be listed in the book stores and will be indexed by DBLP.

II

The main benefit from the change of publisher is that you can read this preface and 22 articles already at the TMRA conference. Furthermore, all articles in this volume will be available online at the conference website without any restrictions. We are convinced, that this decision increases the overall visibility of the work presented at the conference.

The TMRA 2008 program attracted an international crowd from the Topic Maps community, hosted in the media campus of the Leipzig Media Foundation. The scientific quality of the conference was ensured by the international Program Committee with around 50 members. Out of 32 submissions, 22 has been accepted as papers and 5 as posters.

We would like to thank all those who contributed to this book for their excellent work and great cooperation. Furthermore, we want to thank all members of the Program Committee, and especially Prof. Dr. G. Heyer, for their tireless commitment to make TMRA 2008 a success. TMRA was organized by the Zentrum für Informations-, Wissens- und Dienstleistungsmanagement at the University of Leipzig. Furthermore we acknowledge the generous support by all sponsors.

We hope all participants enjoy a successful conference, make a lot of new contacts, gain from fruitful discussions helping to solve current research problems, and have a pleasant stay in Leipzig. Last but not least we hope to see you again at TMRA 2009.

Leipzig and Oslo, October 2008

Lutz Maicher
Lars Marius Garshol

Organization

TMRA 2008 was organized by the Zentrum für Informations-, Wissens- und Dienstleistungsmanagement (ZIWD) in Leipzig, Germany.

Program Committee Chairs

Lutz Maicher, University of Leipzig, DE (Chair)
Lars Marius Garshol, Bouvet, NO (Co-Chair)

Program Committee

Marie-Hélène Abel, University of Technology of Compiègne, FR
Kal Ahmed, NetworkedPlanet, UK
Elham Andaroodi, NII, JP
Frederic Andres, NII, JP
Atta Badii, University of Reading, UK
Xuân Baldauf, University of Auckland, NZ
Robert Barta, rho information systems, AT
Michel Biezunski, Infloom, US
Benjamin Bock, University of Leipzig, DE
Dmitry Bogachev, Omega Business Consulting, CA
Karsten Böhm, FHS Kufstein, AT
Diego Brondo, University of Genoa, IT
Robert Cerny, AT
Michael Chapman, WWW Virtual Library, FR
Iang-Xiang Chen, Yuan Ze University, TW
Darina Dicheva, Winston Salem University, US
Patrick Durusau, US
Nebrasse Ellouze, TN
Andrea Gasparini, University of Oslo Library, NO
Are Gulbrandsen, USIT, University of Oslo, NO
Sung-Kook Han, Won Kwang University, KR
Gerhard Heyer, University of Leipzig, DE
Sachio Hirokawa, Kyushu University, JP
Tobias Hofmann, Bauhaus University Weimar, DE
Gabriel Hopmans, Morpheus, NL
Dongwon Jeong, Kunsan National University, KR

IV

Jirka Kosek, CZ
Marc Wilhelm Küster, FH Worms, DE
Jaeho Lee, University of Seoul, KR
Giovani Rubert Librelotto, UNIFRA - Centro Universtário Franciscano, BR
James David Mason, Y-12 National Security Complex, US
Peter McCarthy, Bibbol, UK
Graham Moore, NetworkedPlanet, UK
Jan Nowitzky, Deutsche Börse Systems, DE
Sam Oh, Sungkyunkwan University, KR
Jack Park, SRI International, US
Rani Pinchuk, Space Application Services, BE
Jörg Schütz, Saarland University, DE
Thomas Schwotzer, FHTW Berlin, DE
Alexander Sigel, University of Cologne, DE
Stefan Smolnik, European Business School, DE
Vaclav Snasel, VSB-Technical University of Ostrava, CZ
Eleni Stroulia, University of Alberta, CA
Volker Stämpflen, Helmholtz Zentrum München, DE
Hendrik Thomas, University of Ilmenau, DE
Kevin Trainor, Ligent, US
Markus Ueberall, University of Frankfurt, DE
Fabio Vitali, University of Bologna, IT
Giuliano Vivinet, University of Genoa, IT

Sponsoring Organizations

NetworkedPlanet, Oxford, UK
Norwegian Computer Society, Oslo, NO
Ravn Webveveriet, Oslo, NO
Media Foundation of the Sparkasse Leipzig, DE
Topic Maps Lab, Leipzig, DE

Program of tutorials@TMRA 2008
Wednesday, October 15, 2008

Full Day (9.00 - 17.00)	Morning Session (9.00 - 12.30)		
<i>Trond Pettersen</i> Room: Schiller Practical Ontology Design for Topic Maps	<i>Lars Heuer</i> Room: Peterhans CTM 1.0	<i>Robert Cerny</i> Room: Everth Topincs	<i>Benjamin Bock</i> Room: Rotunde Introduction to RTM – Ruby Topic Maps
	Afternoon Session (13.30 -17.00)		
	<i>Lars Heuer, Johannes Schmidt</i> Room: Everth TMAPI 2.0	<i>Benjamin Bock</i> Room: Rotunde Fast portal programming with ActiveTM	

Instructor: Trond Pettersen (Bouvet, NO)

Practical Ontology Design for Topic Maps (full day)

Abstract: Designing the ontology is an integral and important aspect of every Topic Maps application. It might sound difficult, but in fact it's not. Learn exactly what an ontology is and how to go about developing one. This day takes the form of an interactive workshop that provides an overview of all the issues to be considered when modeling topic maps. Practical examples are used to apply the methodologies and to help you make modeling decisions based on your requirements. By the conclusion of the tutorial, each participant will have had the opportunity to contribute to the design of a realistic ontology using the methods and principles presented during the workshop.

Audience: This tutorial is suitable for all levels. It is particularly well suited for information architects, CIOs, project managers, system designers and system developers.

Technical Requirements: No technical equipment necessary.

Instructor: Lars Heuer (Semagia, DE)

CTM 1.0 (morning session)

Abstract: CTM is the compact Topic Maps syntax for Topic Maps. This tutorial teaches the language model and syntax. It introduces the basic syntactical constructs and demonstrates their use in a number of examples. Attendees are expected to have a good understanding of the Topic Maps Data Model (TMDM).

Audience: The target audience should be familiar with the Topic Maps Data Model (TMDM), at least with the basics.

Technical Requirements: No technical equipment necessary, but a laptop with a Java Runtime Environment 1.5 (minimum) may be helpful.

Instructor: Robert Cerny (AT)

Topincs - Hands on Topic Maps (morning session)

Abstract: This tutorial shows how to set up and administer a Topic Maps based knowledge repository. It provides a thorough introduction to the software system Topincs which is a server-based Topic Maps solution using the Topic Maps Data Model and REST. The current implementation uses Apache and PHP to serve requests and MySQL for persistence, a simple AMP installation. Topincs consists of two browser-based clients which work on the same data but serve different purposes: (1) Topincs Editor, a topic map editor which offers maximum expressivity and requires in-depth knowledge of the Topic Maps paradigm, and

(2) Topincs Wiki, a semantic wiki for quick editing with limited expressivity for people with intermediate computer skills and little to no knowledge of Topic Maps.

Audience: Anybody interested in agile and distributed knowledge management.

Technical Requirements: A laptop with a recent version of Firefox or Opera. A Topincs installation on the laptop may be helpful, but is not necessary.

Instructor: Benjamin Bock (University of Leipzig, DE)

Introduction to RTM – Ruby Topic Maps (morning session)

Abstract: Ruby Topic Maps (RTM) is a Topic Maps engine created in and for the Ruby programming language. Its focus is an intuitive, easy to use interface, or, as the creators of Ruby would express it: RTM aims to be the Topic Maps programmer's best friend. This tutorial promotes Ruby and RTM to Topic Maps programmers, especially Java programmers who used TMAPI before. The focus of this tutorial is the usage of Ruby and RTM. After a short introduction to Ruby, we'll go on with the usage of the library. We'll look at the usual RTM constructs and highlight major differences to other TM engines.

Audience: The target audience should have a basic knowledge of Topic Maps and programming in general. Ruby skills are not necessary, although beneficial.

Technical Requirements: Each participant should have a laptop with Ruby and Ruby on Rails installed. Ruby is available from <http://www.ruby-lang.org/>, the recommended version is 1.8.6. Ruby on Rails is available from <http://www.rubyonrails.com/>, the recommended version is 2.1.1.

Instructors: Lars Heuer (Semagia, DE),

Johannes Schmidt (Instant Communities GmbH, DE)

TMAPI 2.0 (afternoon session)

Abstract: TMAPI 2.0 is new generation of the common Topic Maps API. This tutorial gives an introduction into the changes between TMAPI 1.0 and 2.0 (Java) and demonstrates the API by several examples. Further, this tutorial will give an outlook how TMAPI was adapted to other programming languages (i.e. PHP5). Attendees are expected to have a good understanding of the Topic Maps Data Model (TMDM) and some experience with Java.

Audience: The target audience has preferable a technical background and is familiar with the Topic Maps - Data Model (TMDM). Knowledge about TMAPI 1.0 is not necessary.

VIII

Technical Requirements: A laptop with a Java Development Kit (JDK) 1.5 (minimum) and an editor (i.e. Eclipse) is preferable. The TMAPI 2.0 library and a TMAPI 2.0 compatible Topic Maps engine will be provided. Participants who would like to experience TMAPI with PHP should have installed an appropriate editor (best: Eclipse with PDT or PHPEclipse). Participants will also need an Apache with PHP5 running and a MySQL-Server version ≥ 5 (means: LAMP/WAMP/MAMP architecture). MySQL access via phpMyAdmin would be great. PHPTMAPI 2.0 library and implementation examples will be provided.

Instructor: Benjamin Bock (University of Leipzig, D)

Fast portal programming with ActiveTM (afternoon session)

Abstract: One of the big applications of Ruby is web development. The flag ship product is Ruby on Rails, a sophisticated web framework optimized for programmer happiness and productivity. With ActiveTM, developers can benefit from Rails and Topic Maps technology at the same time. We'll look at a small sample application using RTM, ActiveTM and Rails and will build our own one.

Audience: The target audience should have a basic knowledge of Topic Maps and web development. Ruby skills are not necessary, although beneficial.

Technical Requirements: Each participant should have a laptop with Ruby and Ruby on Rails installed. Ruby is available from <http://www.ruby-lang.org/>, the recommended version is 1.8.6. Ruby on Rails is available from <http://www.rubyonrails.com/>, the recommended version is 2.1.1. A current version of Netbeans (<http://www.netbeans.org>) is recommended for participants who are not yet familiar with Rails.

Program of TMRA 2008
Thursday, October 16, 2008

09.00 - 9.15 **Welcome Note**

09.15 - 9.30 **Sponsors' Presentation Session**

09.30 - 10.30 *Alexander Johannesen*

We're all crazy - Subjectively speaking

Room: **Schiller**

10.30 - 11.00 **Coffee break**

**Subject-centric
computing**

Room: **Schiller**

11.00 – 12.30 *Gerhard Weber, Ralf
Eilbracht, Stefan Kesberg*

**Topic Maps as Application
Data Model for Subject-
centric Applications**

Jack Park

**Topic Maps, Dashboards
and Sensemaking**

*Robert Barta, Alexander
Zangerl*

**Virtual File System on top
of Topic Maps**

**Topic Maps and
Information Retrieval**

Room: **Everth + Peterhans**

*Markus Ueberall, Oswald
Drobnik*

**Facet-based Exploratory
Search in Topic Maps**

Myongho Yi, Sam Oh

**A Topic map-based ontology
IR system versus
Clustering-based IR System:
A Comparative Study in
Security Domain**

*Roy Lachica, Dino Karabeg,
Sasa Rudan*

**Quality, Relevance and
Importance in Information
Retrieval with Fuzzy
Semantic Networks**

12.30 – 13.30		Lunch Break	
	Standards related research, part 1	Connecting Information	
	Room: Schiller	Room: Everth + Peterhans	
13.30 -14.30	<i>Alexander Mikhailian, Rani Pinchuk, Xuân Baldauf</i>	<i>Thomas Schwotzer</i>	
	A case for XTM 3.0	Building Context Aware P2P Systems with the Shark Framework	
	<i>Hendrik Thomas, Tobias Redmann, Maik Pressler, Bernd Markscheffel</i>	<i>Jörg Wurzer, Stefan Smolnik</i>	
	GTMalpha - Towards a Graphical Notation for Topic Maps	Towards an automatic semantic integration of information	
14.30 - 15.15		Coffee Break	
15.15 - 16.15	Topic Maps Coder Challenge @TMRA 2008		
	Room: Schiller		
	<i>Graham Moore, Marc Wilhelm Küster</i>		
	Protocol for the Syndication of Semantic Descriptions		
16.15 - 16.45		Refreshment	
16.45 - 17.45	<i>Chair: Lars Marius Garshol</i>		
	Open Space Session		
	Room: Schiller		
20.00 - 22.00		Social Event in Café Grundmann	
	August-Bebel-Straße 2, 04275 Leipzig - the most famous Art Deco restaurant in Leipzig -		
	http://www.cafe-grundmann.de/		

**Program of TMRA 2008
Friday, October 17, 2008**

	<i>Towards a new generation of Topic Maps engines</i>	<i>Living Topic Maps</i>
	Room: Schiller	Room: Everth + Peterhans
09.00 – 10.30	<i>Xuân Baldauf, Robert Amor</i>	<i>Martina Husáková, Kamila Olševičová</i>
	Towards a second generation Topic Maps engine	Creating Web Presentation for Observatory and Planetarium with Topic Maps
	<i>Benjamin Bock</i>	<i>Shu Matsuura, Motomu Naito</i>
	ActiveTM - A Topic Maps - Object Mapper	Creating a Topic Maps Based e-Learning System on Introductory Physics
	<i>Lars Heuer</i>	<i>Motomu Naito</i>
	Streaming Topic Maps API	Topic map for Topic Maps case examples
10.30 - 11.30 Coffee Break		
	<i>Standards related research, part 2</i>	<i>Poster Session</i>
	Room: Schiller	Room: Foyer
11.30 – 12.30	<i>Lars Heuer, Johannes Schmidt</i>	<i>Myongho Yi</i>
	TMAPI 2.0	Making Metadata Alive: Migrating Metadata into Richer Semantic Relationships Using Topic Maps-based Ontology
	<i>Lars Marius Garshol</i>	
	TMCL and OWL	<i>Elham Andaroodi, Kinji Ono,</i>

Motomu Naito

**RDF to Topic Map,
Compatibilities and
Differences in Design
Process for Bam 3DCG
Ontology**

Lars Johnsen, Darina Dicheva

**From Connexions Content to
Content Connexions:
Organizing Open Learning
Resources with Topic Maps
and XSLT**

*Adeline Leblanc, Amjad Abou
Assali, Marie-Hélène Abel,
Dominique Lenne*

**Use of Topic Maps to
support Learning
Organizational Memory**

12.30 – 13.30 **Lunch break**

**Topic Maps and Social
Software**

Room: **Schiller**

13.30 -15.00 *Robert Cerny*

**Connecting Topincs - Using
transclusion to connect
proxy spaces**

Sasa Rudan, Sinisa Rudan

**SocioTM – Relevancies,
Collaboration, and Socio-
knowledge in Topic Maps**

Sam Oh, Won Sunmin

Open Space Session

Room: **Everth + Peterhans**

Chair: Lars Marius Garshol

**The Effects of Topic Map
Components on
Serendipitous Information
Retrieval**

15.00 - 15.30 **Coffee Break**

15.30 – 16.15 *Peter Brown*

Everything is subjective

Room: **Schiller**

16.15 - 16.30 **Closing TMRA 2008**

Table of Contents

Subject-centric Computing

Topic Maps as Application Data Model for Subject-centric Applications.....	1
<i>Gerhard E. Weber, Ralf Eilbracht, and Stefan Kesberg</i>	
Topic Maps, Dashboards and Sensemaking.....	11
<i>Jack Park</i>	
Virtual File System on top of Topic Maps.....	31
<i>Alexander Zangerl and Robert Barta</i>	

Topic Maps and Information Retrieval

Facet-based Exploratory Search in Topic Maps.....	49
<i>Markus Ueberall and Oswald Drobnik</i>	
A Topic Maps-based ontology IR system versus Clustering-based IR System: A Comparative Study in Security Domain.....	63
<i>Myongho Yi and Sam Gyun Oh</i>	
Quality, Relevance and Importance in Information Retrieval with Fuzzy Semantic Networks.....	77
<i>Roy Lachica, Dino Karabeg, and Sasha Rudan</i>	

Standards related research

A case for XTM 3.0.....	97
<i>Alexander Mikhailian, Rani Pinchuk, and Xuân Baldauf</i>	
GTMalpha – Towards a Graphical Notation for Topic Maps.....	113
<i>Hendrik Thomas, Tobias Redmann, Maik Pressler, and Bernd Markscheffel</i>	
TMAPI 2.0.....	129
<i>Lars Heuer and Johannes Schmidt</i>	

TMCL and OWL.....	137
<i>Lars Marius Garshol</i>	

Connecting Information

Building Context Aware P2P Systems with the Shark framework.....	157
<i>Thomas Schwotzer</i>	
Towards an automatic semantic integration of information.....	169
<i>Jörg Wurzer and Stefan Smolnik</i>	

Towards a new generation of Topic Maps engines

Towards a second generation Topic Maps engine.....	183
<i>Xuân Baldauf and Robert Amor</i>	
ActiveTM: A Topic Maps – Object Mapper.....	203
<i>Benjamin Bock</i>	
Streaming Topic Maps API.....	219
<i>Lars Heuer</i>	
Protocol for the Syndication of Semantic Descriptions.....	225
<i>Graham Moore and Marc Wilhelm Küster</i>	

Living Topic Maps

Creating Web Presentation for Observatory and Planetarium with Topic Maps.....	237
<i>Martina Husáková and Kamila Olševičová</i>	
Creating a Topic Maps Based e-Learning System on Introductory Physics.....	247
<i>Shu Matsuura and Motomu Naito</i>	
Topic map for Topic Maps case examples.....	261
<i>Motomu Naito</i>	

Topic Maps and Social Software

Connecting Topics Using transclusion to connect proxy spaces.....275
Robert Cerny

SocioTM – Relevancies, Collaboration, and Socio-knowledge
in Topic Maps.....285
Sasha Rudan and Sinisha Rudan

The Effects of Topic Map Components on Serendipitous
Information Retrieval.....301
Sunmin Won and Sam Gyun Oh

Poster Session

The Contributions for the Poster Session.....313
Lutz Maicher and Lars Marius Garshol

Subject-centric Computing

Topic Maps as Application Data Model for Subject-centric Applications

Gerhard E. Weber, Ralf Eilbracht, and Stefan Kesberg

Hoelle & Huettner AG, Tuebingen, Germany¹

Abstract. Today most applications operate on backends with application specific data models. In contrast to this, we suggest modelling application specific information structures at the level of content, and not at the level of the data model. We demonstrate our approach with a publicly accessible web application. Based on a domain ontology, and a set of knowledge models all content for the example application was mapped into a Topic Map. A Topic Maps web frontend renders interface structures, and knowledge-oriented access paths to the highly networked information space of its backend, and also provides relational tables as if it was based on an application specific data model. This approach provides flexible storage layers for applications, and allows using a single data model for different applications. Moreover, applying subject-orientation from high level ontological concepts down to the data level of property values changes accessing content from navigating data-oriented application specific frontends to navigating knowledge-maps.

Keywords: ontology-based application, subject-centric computing, Topic Maps-browser, Topic Maps-frontend, ecotoxicological ontology, Topic Maps-based application.

1. Introduction

Today, most application systems are based on backends with application specific data models, and user interfaces are closely coupled to the relational models of their backends. However, requirements are not static, and very often data models as well as frontends need to be adjusted at high costs. Another disadvantage of application specific data models is the high cost for integrating resources in backends with different models, which is increased by the usually lacking information on the semantics of their constructs. Therefore, we looked for an alternative to building application systems on top of backends with application specific data models and dubious semantics.

¹ Current address: Nexxor GmbH, Stuttgart, Germany; {gerhard.weber, ralf.eilbracht, stefan.kesberg}@nexxor.de

Another option for building application systems would be using a single, and preferably a standardized data model for different applications. Such a data model should enable the attachment of semantics to its constructs. In addition, a standardized processing logic for integrating content in distinct database instances of the data model would provide another benefit with respect to integration requirements. While the Topic Maps data model [6] is aimed at modelling knowledge and connecting encoded knowledge to resources, it is rich enough to be used for modelling knowledge-oriented as well as data-oriented content. It does provide mechanisms for attaching semantics to content and it also defines the sought after processing logic for merging. Therefore, we explored the use of the TMDM [6] as application data model, and demonstrate this approach with a publicly accessible web application developed for the German Federal Environment Agency [11].

2. Using Topic Maps as Application Data Model

The TMDM [6] is a graph-based data model defining a small number of information item types. Out of a total of seven Topic Maps constructs, three fundamental constructs are topics, associations, and occurrences.

A topic is defined as "a symbol within a topic map, which represents a subject about which assertions are to be stated" [6]. Topics may represent "anything whatsoever". That is, topics may represent high level universals like *method*, *object*, *process*, *kind of property* or individuals like a particular sample, a particular property of a particular sample, or even the value of a particular property of a particular sample.

An association is a typed "representation of a relationship between one or more subjects" [6]. The standard defines subject identifiers for the type-instance relationship as well as for the supertype-subtype relationship. With the supertype-subtype relationship hierarchical relationships like the specialisation relationship can be modelled based on standardized subject identifiers. The supertype-subtype relationship is well suited for modelling knowledge-structures such as taxonomies.

An occurrence represents a typed "relationship between a subject and an information resource" [6]. For a topic of type *mean geometric property value*, an occurrence of type *repeatability standard deviation* may be defined, and the value of this occurrence may have a particular data type such as *xs:double*. By providing data types at the level of occurrences, Topic Maps offer an internal bridge over the gap between knowledge-oriented and data-oriented content.

Another Topic Maps construct internally bridging that gap are variant names, the values of which may also have a data type.

The standardized merging process of Topic Maps, supports a stepwise and layered approach for the development of applications. In a first phase a topic map containing the application ontology is created. In a second phase topic maps containing the domain knowledge models are created. In a third phase topic maps with data-oriented content are created. For the productive application all topic maps at the ontology, the knowledge model and the data-oriented level may be merged into a single topic map.

3. An Example for an Application based on Topic Maps

3.1 Background

In the European Union, so far no methodological recommendations for the assessment of the ecotoxicity of waste have been provided. Therefore, the German Federal Environment Agency (FEA) coordinated a European ring test for the evaluation of a battery of methods for assessing the ecotoxicity of wastes and waste eluates described as hazard criterion H14 in the European waste list [[, 9]. The results of this ring test are intended to support the drafting of binding European recommendations for the assessment of the ecotoxicity of waste.

Given the scientific value of the ring test results, and their intended use as expert inputs to strengthen the basis for European environmental policy, all results were to be published online. However, due to the high complexity of the ecotoxicological domain, and the project and data structures, a web-frontend based on an application specific relational data model seemed inappropriate to assure easy accessibility of the expert results for non-experts. Instead, FEA opted for an approach of an entirely Topic Maps-based backend for all information about, and for all results of the ring test. Easy accessibility of the complex data was a prime requirement intended to improve not only expert usage of the results but also the political impact achievable with the outcome of the project².

The ring test focused on three waste substrates which were evaluated by laboratories all over Europe. All in all 67 laboratories participated in the ring test, and 17 different ecotoxicological methods using 16 different biological species were applied. Including the reference substances tested, close to 200 different properties were assessed. The numerical results were subject to statistical

² The H14-Navigator is a commercial customer specific web application created by Hoelle & Huettner AG for the German Federal Environment Agency; to be accessed at <http://EcotoxWasteRingtest.uba.de/h14>.

analysis, which provided the basis for the evaluations of the ecotoxicological methods employed and for the methodological recommendations for assessing ecotoxicity of waste.

3.2 Application Specific Ontology

As an interdisciplinary science, ecotoxicology draws from concepts in a variety of domains such as ecology, biology, chemistry and toxicology. Although, chemical and physical properties of the ring test samples have been excluded from the online publication so far, the core of the ontology was shaped in a way that would allow the mapping of both additional property domains. Therefore, an ontology of physical, chemical and biological properties published by Dybkaer [4] was adapted to the needs of the ring test project and its results.

We aimed for a realistic ontological approach rooted in high level universals such as *object, process, method, property*. For linking these high level universals to the low level universals characterizing the entities dealt with at the laboratory level, we used the superclass-subclass relation.

The adapted ontology on property [4], as well as the project ontology and the ontological components for the biological entities to be modelled was mapped into Topic Maps constructs.

3.3 Integrated Knowledge Models

The sole examination principle used by the ecotoxicological methods studied in the ring test was the response of living organisms exposed to samples of the waste substrates. Considering that taxonomic relations between the species used in the ring test are highly relevant knowledge structures for the test results, a phylogenetic tree was mapped into the Topic Map. It comprised the 16 biological species used in the ring test, and can be considered as one of the core knowledge structures of the ecotoxicological domain. The phylogenetic tree also served for structuring the specialisation hierarchies of the top universals method, process, and property.

3.4 Topic Maps Benefits for Interface Structure and Functionality

All content including results at five different levels from the laboratory level to the level of the ring test recommendations, as well as project structures, the ontology, and all knowledge models were mapped into a single topic map. Due to

the use of association-type names scoped by role-types, and their rendering scoped with the role-type played by the focus topic [8], the well known good readability of binary relations between topics is achieved (Fig. 1).

In the so-called topic view the information displayed for the focused topic is supplemented by visualisations of the relevant knowledge models in order to anchor the subject in the knowledge domain. E.g. the particular Daphnia test displayed in Figure 1 is complemented by the relevant process hierarchy. Thus, even users less familiar with ecotoxicology may recognize that Daphnia tests are ecotoxicological examination processes. Moreover, the process hierarchy offers additional access paths to all instances of its members. Clicking at the high level process type *aquatic ecotox. test* displayed in the hierarchy shown in Fig. 1 would retrieve all tests of this type.

da-w-06
Type(s) : H14RT Daphnia test
Options: Disable tables

Names

da-w-06

Relations (8)

has condition status

- ok

has validity status

- valid

has acceptance status

- accepted

performed by

- Lab 08

used species

- D. magna

performed with

- Vessel

analyzed with

- Probit analysis

used software

- ToxRat

Specialisation

```

graph TD
    Process --> Examination
    Examination --> Ecotox_test[Ecotox. test]
    Ecotox_test --> Aquatic_ecotox_test[Aquatic ecotox. test]
    Aquatic_ecotox_test --> Aquatic_Metazoa_test[Aquatic Metazoa test]
    Aquatic_Metazoa_test --> Daphnia_test[Daphnia test]
    Daphnia_test --> Accute_Daphnia_test[Accute Daphnia test]
    Accute_Daphnia_test --> Daphnia_test_accute[Daphnia test, accute - ISO 6341:1996]
    Daphnia_test_accute --> H14RT_Daphnia_test[H14RT Daphnia test : da-w-06]
                    
```

Test Details

Lab	Method	Sample
Lab 08	H14RT Daphnia method	WOO

Measurements

KindOfProperty	Statistics	Sample	Substance	Value	Unit	ValueType
EC50 D. magna	Probit analysis	WOO		0,19	% dilution	Examined property value
LC50 D. magna			PCD	1,50	mg/kg	Examined reference value

Fig.1: Screenshot detail of a topic view rendered by the H14-Navigator presenting results of a single Daphnia test; the specialisation hierarchy anchors the process in the ecotoxicological domain.

Due to the complexity of the ecotoxicological properties, as well as the ring test structure, views restricted to a single topic, its associations, occurrences, and/or instances are not appropriate to cover all requirements of the application. Relational views are required to visualize results of the ecotoxicological methods used to analyse the three samples. The application specific frontend was therefore enabled to render tables, which provide the same structures as a

frontend using an application specific data model would (Fig. 2). Most cells in these tables do not just contain data of a particular data type but topics representing their subjects of discourse. Users may therefore find more information about any topic displayed, by visualising the linked topic views. E.g. each arithmetic mean value of the sample properties depicted in the table shown in Fig. 2. links to its topic view which visualises its name, published subject indicator, occurrences, associations and associated topics. Thus, the highly networked character of the information related to a mean property value is accessible via the value itself rendered in the user interface.

◊ Sample	◊ Kind of property	◊ Category	Mean arith.	◊ Unit	◊ sr	◊ sR	◊ WL upper
SOI	EC50 B. napus	Effect on terrestrial plants	66,48	% dilution	0,014	0,155	128,32
SOI	EC50 V. fischeri	Effect on aquatic bacteria	64,69	% dilution	0,029	0,094	97,69
SOI	EC50 F. candida reproduction	Effect on terrestrial animals	47,9	% dilution	n.a.	n.a.	n.a.
INC	LC50 E. fetida/andrei EA	Effect on terrestrial animals	46,3	% dilution	0,038	0,086	67,73
INC	EC50 V. fischeri	Effect on aquatic bacteria	40,87	% dilution	0,054	0,254	110,94
SOI	EC50 E. fetida/andrei RA	Effect on terrestrial animals	39,8	% dilution	n.a.	n.a.	n.a.
INC	EC50 E. crypticus reproduction	Effect on terrestrial animals	31,8	% dilution	n.a.	n.a.	58,4
INC	EC50 A. sativa	Effect on terrestrial plants	29,69	% dilution	0,087	0,278	93,29
INC	EC50 F. candida reproduction	Effect on terrestrial	26	% dilution	n.a.	n.a.	n.a.

Fig. 2: Screenshot of the web application H14-Navigator rendering a table with results of the H14 Ring Test; most cells of the table represent topics.

As a further contrast to data-oriented applications, the "data" represented in this knowledge-oriented application are equipped with identities, since published subject identifiers are defined for all topics. The well defined identities of the information items prepare the ground for future integration of content.

Both visualisation patterns – the topic view (cf. Fig. 1), as well as the table view (cf. Fig. 2) – benefit from the integrated knowledge-models. Topic views are

complemented with the relevant knowledge models by a visualisation of complete branches linking specialized concepts with more general concepts. In table views a particular level of a relevant hierarchical knowledge model may be visualized in order to support relating the displayed records in a knowledge domain. E.g. for each record displayed in Fig. 2. a *category* is computed by a recursive inference rule retrieving a particular level in the knowledge model for *kind of property*. The first record depicted in Fig. 2. is thus categorized as quantifying an *effect on terrestrial plants*. In addition, the interface object providing this information represents a topic, and thus offers an access path to either a table of all instances of its kind or its topic view.

4. Discussion

We suggested using the Topic Maps data model [6] as application data model, and demonstrated the approach with a publicly accessible web application [11]. Our method results in benefits at the user interface level, which are due to the differences between a conventional data-oriented approach and the knowledge- or subject-orientation of our approach. Whereas a conventional application would provide a number of data-oriented static tables for accessing content, our approach provides a multitude of access and navigation paths based on a domain ontology, and a number of domain knowledge models. In addition, it also provides relational views on content, analogous to the conventional table views. However, due to the consequent subject-orientation of our approach, most objects rendered in the tables are topics, and not just strings or numbers.

One of the core components of the ontology for our ecotoxicological example is the ontology on physical, chemical and biological properties by Dybkaer [4], which we modified according to the application's requirements and mapped into Topic Maps constructs. Given the effort required for developing an ontology well grounded in a scientific field, it seems worthwhile to tap the wealth of open biomedical ontologies such as described by Smith et al. [10] for the development of knowledge-oriented applications in this domain. Moreover, for publicly visible applications the support of integrative access is close to becoming a requirement. Applications using open topic-mapped ontologies in combination with the advantages Topic Maps offer in terms of integration might therefore offer a promising field for innovative developments.

Our approach of using the TMDM as application data model was first advocated by Ahmed [1] who rightly claimed that the TMDM matches the decomposition of application design into a set of interacting objects. He further stated that using Topic Maps as application data model, would allow modifications of application model structures simply by altering the data which provides the application

schema, thus removing the need to re-compile or re-populate database tables. In short, this comes down to adjusting an application model by changing content of its backend but not structures of its data model and its backend. As a further aspect of considerable advantage Ahmed [1] stated that a single application programming interface would enable accessing the data of any such application.

Although Ahmed [2] elaborated on these ideas, so far Topic Maps has not played a very prominent role as application data model, which might to some degree be due to the slow progress of the standardization process. However, with stable standards for the data model, and the XML syntax [7], with a functional query language [5], and the ISO standardization process for a host of other Topic Maps related standards well under way³, the time for speeding up the exploration of the full potential of Topic Maps technology seems right.

5. Conclusions

We presented an approach for an application architecture which is doing away with application specific data models, as well as with the distinction between data-oriented, and knowledge-oriented content. Applications based on the Topic Maps data model hold the potential for taking subject-centrism into the realm of data, and thus for anchoring subject-oriented computing at the data level. For applications based on this approach, accessing content changes from navigating data-oriented frontends to navigating knowledge-maps.

Acknowledgements

We would like to thank Heidrun Moser of the German Federal Environmental Agency for triggering work on the H14-Navigator, and Jörg Römbke of ECT GmbH for providing ring test contents, and support in ecotoxicological issues.

References

- [1] Ahmed, K.: Topic Maps - A Practical Introduction With Case Studies. XML Europe 2002 (2002)
- [2] Ahmed, K.: Topic Maps for (Open) Source Developers. XTech 2005 (2005)

³ see: <http://www.itscj.ipsj.or.jp/sc34/open/1025.htm>

- [3] Becker, R., Donnevert, G., and Römbke, J.: Biologische Testverfahren zur ökotoxikologischen Charakterisierung von Abfällen - Abschlußbericht. Umweltbundesamt (2008). Available at:
<http://www.umweltdaten.de/publikationen/fpdf-1/3415.pdf>
- [4] Dybkaer, R.: An ontology on property for physical, chemical, and biological systems. APMIS Suppl 117, 1-210 (2004).
- [5] Garshol, L. M.: tolog – A Topic Maps Query Language. TMRA05 International Workshop on Topic Map Research and Applications, L. Maicher and J. Park, eds., pp. 183-196. Springer Berlin, (2006)
- [6] ISO/IEC 13250-2: Information Technology - Topic Maps - Part 2 - Data Model. International Organization for Standardization (2006)
- [7] ISO/IEC 13250-3, Information Technology - Topic Maps - Part 3: XML Syntax. International Organization for Standardization, (2007).
- [8] Pepper, S.: The TAO of topic maps: finding the way in the age of infoglut. XML Europe 2000, (2000). Available at:
<http://www.gca.org/papers/xml europe2000/pdf/s11-01.pdf>
- [9] Römbke, J., Becker, R. and Moser, H. (eds.): Ecotoxicological characterisation of waste - Results and experiences from a European ring test. Springer, New York (in press).
- [10] Smith, B., Ashburner, M., Rosse, C., Bard, J., William, B., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungai, C. J., The OBI Consortium, Leontis, N., Rocca-Serra, P., Rutenberg, A., Sansone, S.-A., Scheuermann, R. H., Shah, N., Whetzel, P., Lewis, S.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology 25, 1251-1255 (2007).
- [11] Weber, G. E., Eilbracht, R. and Kesberg, S.: H14-Navigator uses Topic Maps as application data model. In: Ecotoxicological characterisation of waste - Results and experiences from a European ring test, J. Römbke, R. Becker, and H. Moser (eds.), Springer, New York (in press).

Topic Maps, Dashboards and Sensemaking

Jack Park

SRI International, Menlo Park, CA
and

Knowledge Media Institute, The Open University, Milton Keynes, UK

Abstract. As we migrate from *document-centric* to *subject-centric* computing, we are discovering new approaches to the online facilitation of collective sensemaking. Our approach is to federate with a central topic map the many different tools of hypermedia discourse, such as social bookmarking, semantic annotation, and dialogue mapping. We are learning that this federation facility provides opportunities for unique uses of aggregated sensemaking. We report on our progress in the development of a *dashboard* facility as one such opportunity.

Keywords: sensemaking, hypermedia discourse, topic maps, dashboard

1 Introduction

We first introduced the semantic desktop platform IRIS¹ in 2005 [27]; we described a desktop platform for SRI's Cognitive Assistant that Learns and Organizes (CALO)² project. There, we reported on the need to deal with information overload. In our first report to this workshop [6], we reported on the need to bridge a gap between the work of ontology engineers and that of users of the IRIS platform. In another report to this workshop [3], we discussed the design of a dashboard-like facility for CALO to assist in document preparation. Our work is that of successive refinement of our understandings of knowledge work, now a part of a broader picture referred to as *sensemaking*, making sense of complex issues, and the tools to facilitate sensemaking. We report here on progress made in one aspect of sensemaking: facilitation of document preparation.

¹ IRIS: <http://www.openiris.org/>

² CALO: <http://www.ai.sri.com/project/CALO>

The move from document-centric to subject-centric computing, where everything is a subject [1], is creating new opportunities in sensemaking. Our work combines topic maps with the many tools of hypermedia discourse. Tools like Compendium, which is used in dialogue mapping, and Cohere, which facilitates semantic linking of ideas found on the Web, are part of the hypermedia discourse armamentarium [2]. In our work, we are reporting on essentially the process of rebinding subjects back into documents. It is an interesting, if not ironic process: first, bind subjects into documents by telling stories about those subjects; later harvest those subjects into indexes and topic maps, and later still, rebind them into documents by telling more stories about them. We now are able to report on our progress in the development of dashboard-like capabilities with topic maps. Our report is at once a story of software architectures and of sensemaking processes. Our working hypothesis emerges from the subject-centric nature of topic maps: we can facilitate subject-centric computing by federating (bringing together) the many different representations of world views by re-representing those resources in a uniform frame-like scheme and merging same-subject representations together. Topic maps provide uniform representation of the properties (attributes) of subjects together with the relationships that bind subjects to each other.

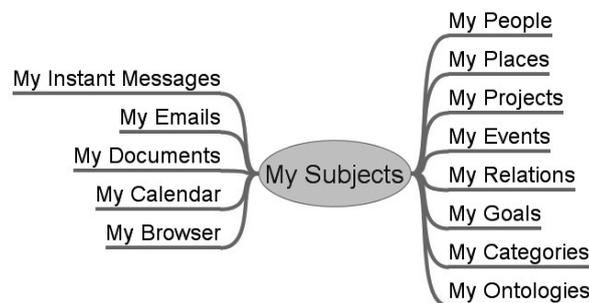


Fig. 1. Topic maps support the migration from documents to subject-centric representation of information resources.

A goal of this report is to describe the tools of sensemaking. We further wish to illustrate the application of those tools in support of dashboard creation and maintenance. Joe Lamantia [9] says that a dashboard “is a portal that combines business intelligence systems and browser-based applications to summarize the status of a complex enterprise for senior decision makers”. We accept that description and extend it to include any user interface element that presents context-appropriate information resources requested by a user. Those resources can include links to related information, paragraphs of related text, images, or other supportive resources.

Our story is about the role of topic maps in the federation of heterogeneous information resources through processes of subject identification and merging different representations of the same subject in the same map (Figure 1). From each document type, we harvest subjects, their attributes and relationships and merge them into a topic map; we thus index and relate the indexed subjects found in the documents. We believe that the maintenance of well-organized information resources can contribute to improvements in collective sensemaking, toward improved human dialogue.

We organize this discussion around sensemaking with a review of three elements of hypermedia discourse that we federate through topic maps. They are social bookmarking, semantic linking, and dialogue (sometimes also known as *issue*) mapping. We close the sensemaking discussion with a review of subject-centric federation processes.

We then sketch architectural aspects of a platform that is designed to provide for the topic maps-based federation. Architectural aspects include client-server capabilities coupled with web services to support the federation of heterogeneous and non-local sensemaking platforms. Our discussion uses TopicSpaces, an independently-developed open source subject map provider we are integrating with the CALO platform. Other topic map platforms can be envisioned to provide the same or similar services. We conclude the paper with an image of a web services provided “bookmark dashboard” that provides links to context-sensitive information resources in a simulated energy sensemaking portal, together with a practical illustration of tagging and semantic linking.

2 Sensemaking

Sensemaking is the social process of *making sense of complex issues and situations*; when facilitated with web-based tools, sensemaking involves elements of hypermedia discourse [2], making use of web-based tools such as blogs, wikis, and tools specific to sensemaking, which we describe here. Following are sections on the three elements of hypermedia discourse being applied to the CALO project. The final section sketches subject-centric *federation*, the process that binds information resources created during sensemaking together with information resources elsewhere on the Web, and the sensemaking process itself.

2.1 Social Bookmarking: Tagging

Tags are associative *reminders*. In the CALO project, tags are the names of projects in which CALO users are engaged. For instance, one typical CALO project is the CALO “platform” itself, a project where CALO developers keep track of the design and development progress on the product. The tag “Platform” would be used by CALO developers as they surf the Web looking for information resources of value to the team. They use that tag with Tagomizer, CALO’s social bookmarking application written on top of the topic map engine TopicSpaces [4], [5].

Tagging is part of the larger social sensemaking repertoire; tags leave *trails* or form *scents* [7] along information foraging [8] paths taken by many. Tagging is part of the *foraging* and *filtering* aspects of sensemaking (see 2.5 below).

While tagging is generally thought to enable the formation of clusters of topics, Brooks and Montanez report some interesting results [24] from experiments with hand-tagged and auto-tagged articles. Using measures of pairwise similarity in the case of human-tagged articles, they conclude that “tagging does manage to group articles into categories, but that there is room for improvement.” They then report on an experiment where they extract, from 500 articles, the three words with the top TFIDF score from each article and use those as “auto tags” for each article. They then cluster the auto-tagged articles. They report better and smaller clusters when compared to human-created tags, and suggest that automated tagging can add great value to search for topics using tags.

2.2 Semantic Linking

In some sense the entire Semantic Web enterprise is about semantic linking. In the sense discussed here, a narrow definition is taken: semantic linking here refers to the creation of typed connections between *ideas* found in documents on the web. In that sense, semantic linking is subject-centric by its very nature. In 2001, the Scholarly Ontologies Project at the Knowledge Media Institute began to envision a “complementary infrastructure that is ‘native’ to the Internet, enabling more effective dissemination, debate, and analysis of ideas”³. In 1999, three authors [10] proposed that when a new article is to be published, “authors describe the document’s main contributions and relationships to the literature using a controlled vocabulary analogous to a metadata scheme (but implemented using a formal ontology), and submit the description to a networked repository.” In more recent writing [11], the Cohere project (Figure 2) has been described as

³ ScholOnto: <http://kmi.open.ac.uk/projects/scholonto/>

an online means where social processes are used to find and annotate ideas on the Web.

Fig. 2. Cohere⁴ Semantic Linking Web Portal

2.3 Dialogue Mapping

Dialogue mapping provides a common view of a growing structured representation of streams of thoughts [12]. In fact, there are limits to conversation [13]. Starting with a linear collection of thoughts, it is possible to tease out of that collection a starting question followed by statements that answer the question, statements that argue about the answers, and possibly statements that raise new questions.

Analyzing a large body of text into such a map is called *issue mapping*⁵. For instance, a recent *OpEd* discussion⁶ about *food riots* was mapped by the author as illustrated in Figure 3.

⁴ Cohere: <http://cohere.open.ac.uk/>

⁵ Issue mapping: http://cognexus.org/issue_mapping.htm

⁶ OpEd: <http://www.nytimes.com/2008/04/07/opinion/07krugman.html>

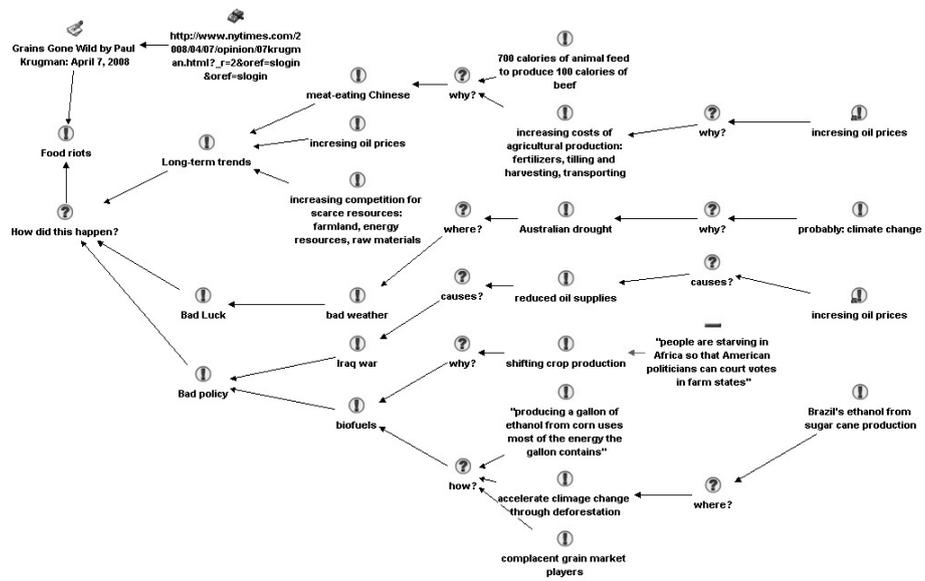


Fig. 3. Finding structure in an OpEd with Compendium.

The map reads left to right, starting, essentially, with an opening question. The node “Food riots” leads to the columnist’s opening question: “How did this happen?” The columnist provided his own three answers: “Long term trends”, “Bad luck”, and “Bad policy”. From there, it is a matter of picking out questions being asked, finding answers and identifying any arguments made in the prose. A similar dialogue map would occur if a discussion group was facilitated by a skilled dialogue mapper and similar questions and responses were recorded.

2.3.1 Related work

Tools that support dialogue mapping include Compendium⁷, B-Cisive⁸, TruthMapping⁹, and DebateGraph¹⁰. Compendium and B-Cisive are desktop tools, whereas TruthMapping and DebateGraph are online portals.

Mark Klein [16] describes online dialogue mapping on a large scale. He describes the popular communications tools, instant messaging, email, forums and wikis, as facing “serious shortcomings from the standpoint of enhancing

⁷ Compendium: <http://compendium.open.ac.uk/>
⁸ B-Cisive: <http://bcisive.austhink.com/>
⁹ TruthMapping: <http://truthmapping.com/>
¹⁰ DebateGraph: <http://debategraph.org/>

collective intelligence”. He then goes on to describe the need for maintaining structure in conversations as was discussed above in Section 2.3.

2.4 Subject-centric Federation

We live in a vast collection of universes of discourse, each centered in different topic domains, many of which overlap and share subjects and concerns. The issue map of the OpEd illustrated above could just as easily have been generated in slightly different forms, each representing a different interpretation by a different analyst. That each is somehow different contributes to heterogeneity in information resources with which we must all cope in our day-to-day and decision-making lives. A goal of our work is to *federate* these heterogeneous resources into a coherent representation with which we believe improved collective sensemaking is afforded.

Consider just one node in our OpEd issue map (Figure 4), the one for which the label reads “700 calories of animal feed to produce 100 calories of beef”. That is a specific quote from the OpEd text; it is reasonable to expect that other analysts might pick up the same *claim*, even if placed in a different part of their map’s graph structure.



Fig. 4. A *Claim* found in the OpEd and represented in the issue map.

Claims such as that are, at once, subject to fact checking, and to *entailed subjects*. Fact checking can be the work of background agents, or the work of the *crowd* engaged in social sensemaking. Subject entailment goes with the nature of the claim. That is, there is a relationship between animal feed and animals, and both of those two subjects exist in a web of related (entailed) subjects. Consider the simple concept map in Figure 5 of some (but not all) subjects entailed by the node illustrated above.

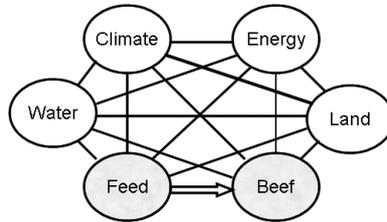


Fig. 5. Subjects entailed by the two subjects “Feed” and “Beef”.

By creating a topic map of dialogues, and by including all entailed subjects, we gain a broader means by which the work products of collective sensemaking can be evaluated. By linking into that map each node created by each individual, no matter how that node falls in its native dialogue map structure, we are performing subject-centric federation: we are *bringing together* information resources that are *about* the same subject, and we are connecting those resources to all known-to-the-map resources of the same or related subjects. We do so without editorial bias; we federate regardless of whether or not we agree with claims represented. We leave disagreements to the collective sensemaking processes in which the map’s users are engaged.

2.5 Sensemaking Processes

As we continue to evolve our tools, and as we use them in our own research, we are beginning to understand, if even to a somewhat naïve level, what so-called - *best practices* might look like. We now understand some best practices for tagging, and are just now beginning to practice semantic linking and dialogue mapping. Those best practices exist in the context of the larger *sensemaking* process. The sensemaking process occurs within some context, some goal, some working hypothesis or research question. Our ideas draw from other sensemaking research as described below (2.5.1).

We see the sensemaking process as iteration around and within this sequence:

1. Forage
2. Filter
3. Analyze
4. Synthesize

Foraging and filtering are the information-seeking stages in which combinations of goal-directed search and thematic vagabonding result in discovered

information resources. In this stage, one tags the resources for later harvesting. This is the stage where benefits accrue from tagging best practices. In our CALO scenario above, we described the application of a *project-centric* tag ontology, the use of predefined tags for specific purposes. We are learning that it is appropriate to use more than one tag for each resource discovered. While CALO prescribed *project-centric* tags, we further prescribe *subject-centric* tags. While reading a particular resource on discovery, take the time to tag the particular actors, relationships, states and other important subjects bound by the resource. This extra work pays large dividends later.

Concurrent with tagging, semantic linking serves as a transition to analysis through partial harvesting and forging semantic connections between ideas harvested from the pages visited. We are able to use the full suite of hypermedia discourse tools in the foraging-filtering stages and in transitioning to analysis [17] and [18]. Figure 8 illustrates how we used Compendium, with a simulated Cohere connection to organize a literature search related to subject identity.

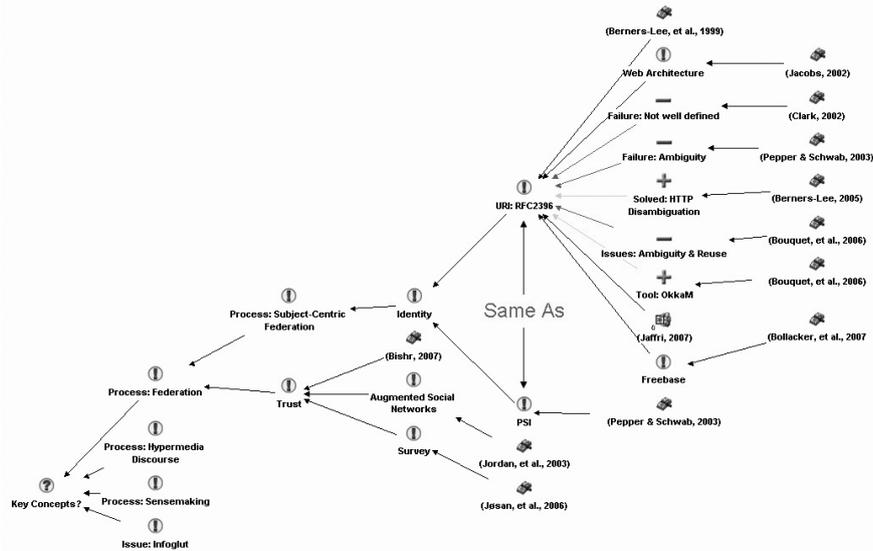


Fig. 6. Using Compendium and Cohere (simulated) to organize a literature review.

Reading this issue map from left to right, it organizes the concepts about which our literature review must speak. Toward the right, we begin to tease out of the literature each argument made, and we tie each argument to the specific citation from which it is drawn. The two key concepts were URIs from the Web community and PSIs from the topic maps community. Our analysis suggests that they behave as the same concept, and we note that through a Cohere-like coherence relation.

The analysis stage includes finding answers to research questions posed at the beginning of the process, and derives new questions to ask and finds their answers—or reports them as targets for future work. In the analysis stage, some assertions made during foraging and filtering, our *Same As* assertion, for instance, may come under close scrutiny by those who do not share the same world views. This is the point where dialogue mapping services enter the arena and various participants take positions and offer arguments—sensemaking at work.

2.5.1 Related work

Brenda Dervin's sensemaking methodology [20] is characterized as bridging a situation-outcome gap. A visual imagination suggests similarity to Gowin's Vee, as sketched Figure 7, [19] where her situation is modeled as the present state of a learner in terms of conceptual knowledge, the outcome is modeled as the work product of performance, and the gap represents question answering and feedback. Gowin's Vee diagram illustrates the processes of constructivist learning where a focus concept provokes questions for which the learner, applying existing personal knowledge, articulates answers, writes reports, and engages in responding to feedback.

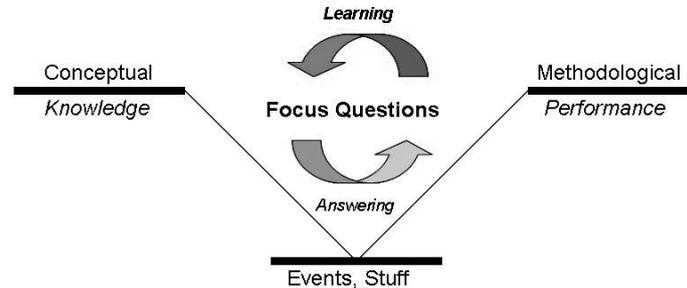


Fig. 7. Gowin's Vee (after [19])

Sensemaking has been approached from the perspective of surprise, of *expectation failures* [22]. Sensemaking is defined [22] as the deliberate effort to understand events and is typically triggered by unexpected changes or surprises that make a decision maker doubt prior understanding. The authors [22] further characterize the process as active, building, refining, questioning, and recovering situation awareness. Elements of their "Data-Frame Model of Sensemaking" sketched in their paper are these:

- Recognize and construct a frame

- Perform cycles of elaboration on that frame, adding and filling slots, seeking, inferring and discovering data
- Ask questions of the frame, detecting inconsistencies, judging plausibility, analyzing data quality
- Perform cycles of refactoring, where the process is to seek a new frame that better describes the situation

In the *line of inquiry* framework [21], the sensemaking is facilitated by a framework that embodies theories, questions, information seeking strategies, evidence and evidence collections, knowledge, assigned investigators, and lower-level lines of inquiry. As suggested in the paper's title, this is a recursive framework. A line of inquiry will spawn subinquiries, each of which is treated as a fully embodied line of inquiry. Elements of the framework are

- Generate theories
- Ask questions
- Seek new information
- Collect evidence
- Gain new knowledge
- Assign investigators
- Spawn subinquiries

Jean-Claude Bradley [23] describes a generalized sensemaking process he calls *Open Notebook Science*. He coined the term to avoid ambiguities associated with the name *Open Source Science*. He describes a process wherein a traditional *lab notebook* is implemented with a wiki platform, and blog entries are used to tell stories about events and findings described in the notebook.

Standing by itself as a new class of sensemaking portal is Science X2¹¹. The portal provides users with dashboards that consist of unread posts to groups to which the user is subscribed, lists of “signals”, “hypotheses”, and “forecasts” generated by the user. While we have only a “beginner’s” experience with the Web site, it appears that users post *signals*, an instance of which might be “Topic maps improve sensemaking”, and other users form *hypotheses* around such signals and later offer *forecasts*. We view this portal as federation of goal-oriented blogs, tightly coupled through the three classes of artifacts. In some sense, the portal, by virtue of its three specified artifacts, is naturally self-organizing in a subject-centric fashion.

¹¹ Science X2: <http://sciencex2.org/>

3 A platform for sensemaking

We describe TopicSpaces, the experimental platform for sensemaking we applied to our CALO projects. TopicSpaces is a servlet-based Web portal provider that includes a subject map, which is a topic map created according to the Topic Maps Reference Model [14]. The platform provides a servlet-driven REST API [25] for Web services, and will later provide a tuplespace agent coordination platform [15] to coordinate harvesting agents on the Web and those included in desktop applications.

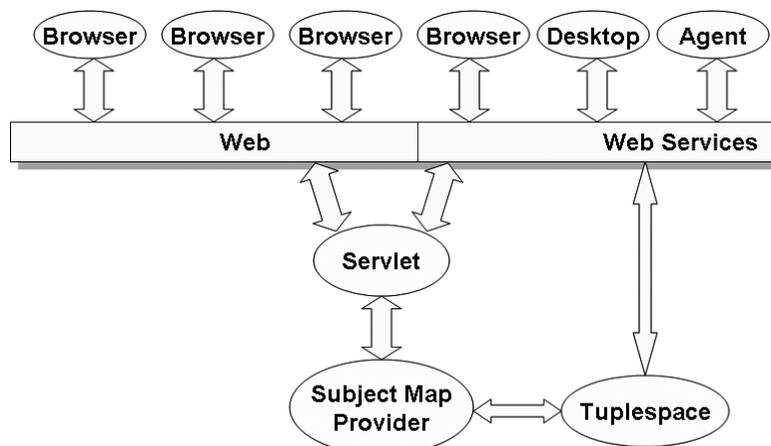


Fig. 8. The TopicSpaces Platform Architecture

The platform illustrated in Figure 8 anticipates the ability to run *seti@home*-like agent-based harvesting of resources found on the Web. For instance, consider the scenario where a user tags a website that is new to the TopicSpaces portal. That new resource is sent to a harvesting agent that can either perform harvesting tasks locally, or post a new harvesting task to the Tuplespace where agents elsewhere on the Internet have authenticated and are waiting for harvesting tasks. A typical harvesting task, well suited to topic-mapped resources is that of the TextRunner process [26], where bodies of text are parsed, not for sentence structure, but for noun and verb phrases from which concept maps are constructed that represent the material being “read” by the agent. The TextRunner approach parses bodies of text into lists of triples of type $\{entity_i, relation_{i,j}, entity_j\}$ from which concept maps, later topic maps, can be constructed. We believe that the topic map’s attention to the details of subject identity can render this process more accurate; to do so, an iterative process of

comparison of the resulting concept maps with their corresponding named topics in a topic map will allow refinement of the concept map before migrating it into the topic map. This will be particularly important in cases where named concepts found by the TextRunner algorithm are determined to be ambiguous; different entities with the same name create such ambiguities.

3.1 Portals

TopicSpaces can support two classes of topic maps portals as illustrated in Figure 9. For one class of portal, the *all-in-one*, all of the context view portals, collaboration portals, and personal workspaces are part of the same software package. TopicSpaces is built like that as a means to explore all issues related to sensemaking.

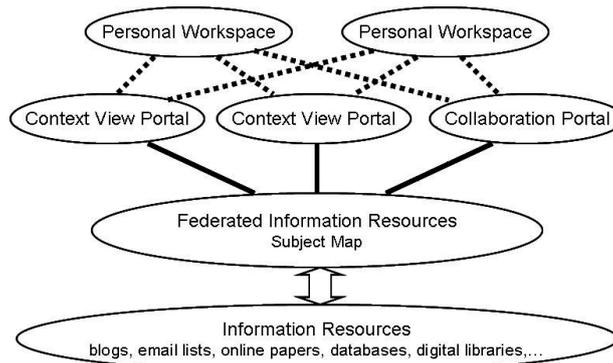


Fig. 9. The TopicSpaces Web Portal Architecture

A second class of portal separates all the context portals, collaboration portals, and so forth from the subject map itself. Different portals can then be crafted using standard CMS platforms such as Drupal, WordPress, and other popular software products. TopicSpaces can provide Web services to those portals as needed.

3.2 REST Web Services API

What is a REST Web service? It is simply a means to use URLs as query vehicles by way of a servlet. Web browsers make such requests routinely; type a particular URL into a browser and the server returns the entire Web page in a

single HTML string. A web service would, instead, return a small fragment of HTML, of XML, or Javascript Object Notation (JSON)¹² as requested. Bookmarklets, as used by Tagomizer, del.icio.us, and other social Web sites, represent a kind of web service where a short javascript string embedded in a browser's bookmarks is able to transport information from a Web page to the portal that accepts the Bookmarklet's query. When we say "API", we are specifying that there is a particular *query string* that goes in the URL, and that query string is interpreted by the portal to perform the requested task. Some tasks are to return a requested bit of information, the bookmarks associated with a particular tag, say. Other tasks are to update information in the topic map, to add a new bookmark, say.

The TopicSpaces REST API takes the form:

```
<server>/ws/<appname>/<object>/<return>/<data>
```

For instance, asking for a Tagomizer tag in HTML where the tag is "SomeTag" is this query fragment:

```
/ws/tago/tag/html/SomeTag
```

The same query returning the result in JSON is this:

```
/ws/tago/tag/json/SomeTag
```

3.3 Related work

Platforms closest to this discussion are *Fuzzy*¹³ [28], a social bookmarking site built on a topic map platform, NexistWiki [29], a wiki and topic map combination, used in the Bay Area Science Collaboratory¹⁴ education project, and the Hypertopic project [20]. Hypertopic is the name of a topic map architecture that federates multiple topic maps, each expressing a different point of view.

4 Discussion

A general outcome of this work for CALO is to provide a Web-based presence that supports collective sensemaking among communities of CALO users, as illustrated in Figure 10.

¹² JSON: <http://www.json.org/>

¹³ Fuzzy: <http://fuzzy.com/>

¹⁴ Nexist Wiki: <http://www.nexist.org/hf/>

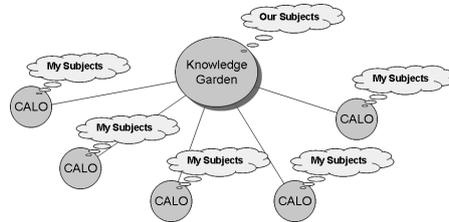


Fig. 10. Topic map-based sensemaking portal¹⁵ for communities of CALO users.

In the work reported here, we have installed an instance of TopicSpaces and we have begun to use it in two different contexts: developing a dashboard platform for CALO, and using it to organize our thesis research, snippets of which have been illustrated here. An allied goal has been to demonstrate the ability to federate communities of CALO users where subjects important to all members of the community are shared and maintained at the Web portal.

Energy Sources

Sources	Bookmarks
Solar Energy	Solar energy - Wikipedia, the free encyclopedia View Bookmark in Wiki Monday, March 24, 2008 7:31:26 PM PDT by jackpark under GENIS:Source, 1 bookmarks for this resource
Wind Energy	American Wind Energy Association View Bookmark in Wiki Monday, March 24, 2008 7:33:57 PM PDT by jackpark under GENIS:Source, association, 1 bookmarks for this resource
	IEEE Spectrum: Super Soaker Inventor Invents New Thermoelectric Generator View Bookmark in Wiki Friday, March 28, 2008 7:12:21 AM PDT by jackpark under GENIS:Source, solar thermal to electricity, 1 bookmarks for this resource
	Solar Thermal Electricity: Can it Replace Coal, Gas, and Oil? : CleanTechnica View Bookmark in Wiki Friday, March 28, 2008 8:31:51 AM PDT by jackpark under GENIS:Source, solar thermal to electricity, GENIS:Issue, 1 bookmarks for this resource
	First Algae Biodiesel Plant Goes Online: April 1, 2008 : Gas 2.0 View Bookmark in Wiki Saturday, March 29, 2008 6:53:04 PM PDT by jackpark under GENIS:Source, energy, biofuel, algae, 1 bookmarks for this resource

Fig. 11. A sample Portal with a bookmarks Dashboard view.

Figure 11 illustrates an early instance of a dashboard, in this case, a simple text widget in the right half of the pane that displays related bookmarks. This dashboard uses a REST query as follows:

¹⁵ Knowledge Garden: a name we have given to the TopicSpaces platform

```
/ws/tago/tag/html/GENIS:Source
```

where the tag `GENIS:Source` is drawn from a tag ontology that allows us to bookmark using tags related to energy sources, uses, and issues.

Figure 11 also illustrates a *context view portal* (Figure 8), where we have created a view that facilitates navigation into the world of *energy sources*. Users are able to create new source links, and are also given ready access to Web sites tagged for the general class. A Web page for the subject *Wind Energy* (source) might include a bookmark dashboard that is a composite query on `GENIS:Source + WindPower`, which narrows the source bookmarks to those also tagged with the particular source type.

Through such tagging and annotating processes, we believe that it is possible for communities of practice to create and maintain a knowledge base that fully supports the community's sensemaking activities through maintenance of dashboards of various kinds. Dialogue mapping provides a means by which the *gap-bridging* of Dervin's methodology is facilitated. Through dialogues, it is possible for a structured discussion to reveal concepts and related ideas that bridge some gap, answer some question that went unanswered before.

Tagging provides a means to couple concepts together, while semantically-linked ideas lifted out of Web pages wire a web of ideas together in coherent ways. Consider an example consisting of two different Web pages. One page contains a story from which we lift the following idea related to immune system behaviors:

“macrophages generate freeradical molecules to fight bacteria”

A second Web page contains a story from which we lift the following idea related to antioxidant supplement pills:

“antioxidants kill freeradical molecules”

Suppose now that we happened to tag each page with the tag “freeradical molecules”. Allow that different individuals used the same tag and lifted the ideas independently.

Later, someone performs a tag-based search or discovers the tag “freeradical molecules” and notices those two ideas together. Those two ideas, to astute viewers, pose a problem: if you need freeradical molecules to fight bacterial infections, you probably don't want to be taking high-dose antioxidants. Those two ideas should be semantically linked with a relation type that expresses that issue. Perhaps that new relationship can be made to appear in a health-care related context where a user should be warned of the issue through a dashboard that expresses necessary warnings.

Our work remains *in progress* as we continue to explore the boundaries of dashboard construction; we have only now begun to scratch the surface of that inquiry.

Acknowledgments. This material is based in part upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. 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 DARPA, or the Air Force Research Laboratory (AFRL).

References

1. Pepper, Steve (2008). "Everything is a Subject". *Topic Maps 2008*, Oslo, Norway 2-4 April.
2. Buckingham Shum, S. (2006). "Sensemaking on the Pragmatic Web: A Hypermedia Discourse Perspective". Proc. *PragWeb'06: 1st International Conference on the Pragmatic Web*, Stuttgart, 21-23 Sept.
3. Park, Jack (2007). "Towards a Topic Maps Amanuensis". *TMRA 07*. Leipzig, Germany, 6-8 October.
4. Park, Jack (2006). "Tagomizer: Subject Maps Meet Social Bookmarking". *TMRA 06*. Leipzig, Germany, 5-7 October.
5. Park, Jack (2006). "Promiscuous Semantic Federation. Semantic Desktops Meet Web 2.0". *Semantic Desktop Workshop 2006*. Athens, GA, 6 November.
6. Park, Jack, and Adam Cheyer (2005). "Just For Me: Topic Maps and Ontologies". *TMRA 05*. Leipzig, Germany. 6-7 October.
7. Chi, Ed H., Peter Pirolli, and James Pitkow (2001). "'The Scent of a Site: A System for Analyzing and Predicting Information Scent, Usage, and Usability of a Web Site". In *Proc. of ACM CHI 2000 Conference on Human Factors in Computing Systems*, 161-168.
8. Chi, Ed H., Peter Pirolli, and Shyong K. Lam (2007). "Aspects of Augmented Social Cognition: Social Information Foraging and Social Search". LNCS 4564, 60-69.
9. Lamantia, Joe (2006). "The Challenge of Dashboards and Portals". Weblog entry online at <http://www.boxesandarrows.com/view/the-challenge-of>
10. Buckingham Shum, Simon, Enrico Motta, and John Domingue (1999) "Representing Scholarly Claims in Internet Digital Libraries: A Knowledge Modelling Approach". LNCS 1696, 423-442
11. Buckingham Shum, S. (2008a). "Cohere: Towards Web 2.0 Argumentation". Proc. *COMMA'08: 2nd International Conference on Computational Models of Argument*, 28-30 May 2008, Toulouse. IOS Press: Amsterdam

12. Conklin, Jeff (2005). *Dialogue Mapping: Building Shared Understanding of Wicked Problems*. Wiley.
13. Conklin, Jeff (2008). "Limits of Conversation Structure". Screencast online at <http://www.youtube.com/watch?v=pxS5wUljfjE> and at http://cognexus.org/videos/ConversationVsIssueStructure_V4/ConversationVsIssueStructure%20V4.html
14. ISO (2005) "ISO/IEC CD 13250-5 Topic Maps — Part 5: Reference Model". Online at <http://www.isotopicmaps.org/TMRM/TMRM-5.0/TMRM-5.0.pdf>
15. Carriero, Nicholas, and David Gelernter (1989). "Linda in Context". *Communications of the ACM*, April 1989, Vol. 32, No. 4, pp. 444-458
16. Klein, Mark (2007). "Achieving Collective Intelligence Via Large-Scale On-Line Argumentation" (April 1, 2007). MIT Sloan Research Paper No. 4647-07 Online at <http://ssrn.com/abstract=1040881>
17. Selvin, Albert M. (1999). "Supporting Collaborative Analysis and Design with Hypertext Functionality". *JODI* Volume 1, Issue 4 Article No. 16, 1999-01-14
18. Okada, A. and Buckingham Shum, S. (2006). "Knowledge Mapping with Compendium in Academic Research and Online Education". *22nd ICDE World Conference*, 3-6 Sept. 2006
19. Novak, J. D., and D. B. Gowin (1984). *Learning How to Learn*. Cambridge, UK: Cambridge University Press
20. Naumer, Charles M., Karen E. Fisher, and Brenda Dervin (2008). "Sense-Making: A Methodological Perspective". *Sensemaking Workshop, CHI'08*. Florence, Italy. April, 2008.
21. Attfeld, Simon, Ann Blandford, and Stephen De Gabrielle (2008). "Investigations within Investigations: A Recursive Framework for Scalable Sensemaking Support". *Sensemaking Workshop, CHI'08*. Florence, Italy. April, 2008.
22. Hutton, Robert, Gary Klein, and Sterling Wiggins (2008). "Designing for Sensemaking: A Macrocognitive Approach". *Sensemaking Workshop, CHI'08*. Florence, Italy. April, 2008.
23. Bradley, Jean-Claude (2008). "Open Notebook Science: Implications for the Future of Libraries", slide presentation at the University of British Columbia School of Library, Archival and Information Studies (SLAIS), April 2, 2008.
24. Brooks, C. H., and N. Montanez (2006). "Improved annotation of the blogosphere via autotagging and hierarchical clustering". In *Proceedings of the 15th International Conference on World Wide Web* (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 625-632.
25. Fielding, Roy Thomas (2000). "Architectural Styles and the Design of Network-based Software Architectures". Ph.D. dissertation. UCI. Online at <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
26. Banko, Michele, Michael J. Cafarella, Stephen Soderland, Matt Brodhead, and Oren Etzioni (2007). "Open Information Extraction from the Web". *IJCAI 2007*. Hyderabad, India. January, 2007.

27. Cheyer, Adam, Jack Park, and Richard Giuli, "IRIS: Integrate. Relate. Infer. Share," 1st International Workshop on The Semantic Desktop, Galway, Ireland, 6 November 2005.
28. Karabeg, Dino, and Roy Lachica (2007). "Towards holistic knowledge creation and interchange. Part II: Examples, theory and strategy." *Third International Conference on Topic Maps Research and Applications: TMRA 2007*.
29. Kahn, Ted (2007). "Science Museum Learning Collaboratories: Helping to Bridge the Gap Between Museums' Informal Learning Resources and Science Education in K-12 Schools", *Museums and the Web 2007*, April 11-14, 2007, San Francisco, California.
30. Cahier, Jean-Pierre, and Manuel Zacklad (2004). "'Socio-Semantic Web' applications: towards a methodology based on the Theory of the Communities of Action". COOP'04 Workshop on Knowledge Interaction and Knowledge Management. French Riviera - May 11, 2004.

Virtual File System on top of Topic Maps

Alexander Zangerl¹ and Robert Barta²

¹Bond University, School of Information Technology

²Austrian Research Centers Seibersdorf

az@bond.edu.au

robert.barta@arcs.ac.at

Abstract. The UNIX file system provides a robust framework to abstract away from technical differences between various storage media. This work summarizes experiences to define a virtual file system on top of an existing topic map. It clarifies the involved concepts, details the mapping and reports on first experiences with a proof-of-concept implementation.

1 Introduction

The UNIX file system paradigm has proven to be a flexible - while relatively efficient – abstraction layer between applications and underlying storage media. Accordingly, the operating system offers access not only to data on disk, but also to other media by organizing byte sequences along a tree structure of directories. As an escape hatch, mainly to break up the rigid tree-structure, the UNIX file system provides additionally uni-directional links from one file to another.

In the past, numerous storage technologies have been successfully implemented, differing in disk usage profile, speed, reliability, etc. The mechanism itself has been extended to pseudo file systems, such as for instance `/proc` or `udev` in Linux kernels, which export process or device characteristics in form of a file-based directory hierarchy.

In this sense a file system is always virtual; applications which follow that ontological commitment of files and directories always have to open files, read and/or write their content and close them afterwards, oblivious of the underlying storage technology and its idiosyncratic consistency rules. Notable examples of such file systems include SSHFS [12] which provides access to a remote file system through an SSH tunnel, and GmailFS [9], which makes a Google GMail mail storage account accessible for general-purpose data storage.

Also semantic networks have already been brought into this scheme, in particular using RDF [11] [10]. As the RDF model offers only the construct of a triple, such mapping is rather straightforward. The flatness of the model makes it less useful for humans, though. Using Topic Maps as underlying storage technology is motivated by a number of factors:

- Such an abstraction couches Topic Map constructs in terms of well-known first-class objects such as directories and files. This can lower the intellectual investment necessary to adopt Topic Maps technologies into applications.
- Using files and directories as abstraction layer can dramatically simplify the day-to-day handling of manually managed Topic Maps content. Writing text strings into files is significantly easier and faster for a human user than using a dedicated TM application.
- Navigation through a topic map is translated to navigation through the file system, something which typical command line-oriented interpreters are particularly good at. The same applies to simple lookups into the map (such as *what is the Wikipedia page for concept X*) or simple bulk operations (*give me everything you know about concept X*).
- With a transition from Topic Maps to a file system also the access granularity is changed. Whereas a TM application will have to manage a whole map, broken down to various information items, any access mechanism via a file system has to use a much smaller number of basic concepts.
- On a technological level applications become so independent from any Topic Map library and are effectively TM-API agnostic; and they will continue to operate if the file system uses a completely different technology at some later stage.

The TM file system (TMFS) proposed here covers (a) the translation of TM items belonging to a single map onto a single file system that follows the UNIX file model. This is (b) extended to a translation involving a whole hierarchy of maps.

First it is necessary to clarify the concepts concerning the UNIX file system (section 2). Assuming the reader is familiar with Topic Maps concepts [2] the bidirectional translation between a topic map instance and a single file system is presented in section 3. In a second step *map spheres* [4] [5] are reinterpreted as file systems. In section 4 the practicability of the mapping is demonstrated, along a number of use cases which influenced our design.

To test the approach for feasibility a proof-of-concept implementation based on an existing TM framework has been created. In section 5 some interesting

aspects are elaborated on, so is the list of current deficiencies. This is complemented by discussions regarding the design, scalability and future work (sections 6 to 7).

2 UNIX File System

File systems conforming to the UNIX file model [1] provide a number of mechanisms for organizing content. All are strictly hierarchical and are organized into a single tree of directories. Each directory can contain other directories, (regular) files and a number of special file-like objects:

- *Regular Files* are represented in a file system by a node with a unique internal identifier. Additionally files have a name whereby there are only a few limitations concerning the allowed character set. Every file holds content in form of streams. Specialized files contain only alphanumeric (text) content.
- *Directories* are containers of nodes for files or other directories. Two files/directories must differ in name to make them unique within the directory. Also directories have names and they also have a reference to the parent directory they are in.
- *Symbolic Links* are files in their own right. They are a reference of one file to another (with some limitations) and possibly can also point to a non-existing file. File resolution is normally transparent for applications but they can test files for being symlinks.
- *Mounting* is the process of attaching one storage mechanism to one subtree within a directory structure. All subsequent accesses to that directory or areas below the mount point are translated into accesses to the newly mounted device.

A name within a particular directory can reference a single object only. A named object is either a directory, or a file, be that regular or special. This will guide the conceptual translation of file accesses into the virtual file system to the navigation within a topic map.

Given these types of objects the semantics is determined by operations such as `mkdir` (to create a directory), `rmdir` (to remove it), `open`, `read` and `close` to open, read and close files, and so forth. There are about 2 dozens of these operations, although most of the common functionality is concentrated in a handful of them.

3 Topic Maps to File System Mapping

The design choices for translating a topic map structure into a tree structure were mostly driven by user convenience, not so much by functional completeness. Particular care was taken that the mapping also considers the copious use of text-based notations (such as CTM [8]). This mix opens the pathway to reuse the large existing tool-base for text processing under UNIX (such as `grep`, `find`, etc.).

The mapping between the file system tree and Topic Map content is done on two levels. Whole subtrees in a file system can be directly translated into a *map sphere* [5], i.e. a structure which generalizes a single topic map into one which can contain submaps. Individual maps are mapped into one directory.

3.1 Map Spheres

Map spheres are a generalization of a standard Topic Maps structure. Their purpose is to host not only TM content, but also topics of a predefined type which reify further topic maps. As these topic maps can in turn contain further submaps the overall structure takes the form of a tree. To provide a convenient addressing scheme, we adopt a tree-notation of the form `/markup/xml/xslt/`. According to the definition of a map sphere the above expression would locate the top-level map first and would find a topic `markup` in there. That topic is supposed to reify a whole map, also stored inside the map sphere. In that map the topic with the identifier `xml` is expected. With it the same procedure will repeat, as will with `xslt`. At the end of this process a whole map is addressed.

As the processes of *attaching one map* into another is quite similar to *mounting* of file systems, the choice of notation was already inspired by file system navigation.

3.2 Maps

One individual map is represented in the virtual file system as directory. Individual topics will be direct subdirectories thereof, whereby we use topic item identifiers as names for the directory.

For associations we choose a different route in that there exists one directory `.associations` containing all of them. Additionally to associations two more directories are provided: One called `.characteristics`, which contains all names and occurrences, and a second, `.assertions` which contains all of the

above. In all cases an internal association identifier is used as name for a directory.

Apart of these subdirectories the map directory also contains additional *dot* files, such as `.ctm` or `.xtm`. When accessed, they return the map in the respective serialization format. The naming choice with a leading `.` is deliberate: for many UNIX tools it is a convention to treat such files as *hidden*.

3.3 Topics

Internal topic identifiers are quite convenient to be used as basis for naming the directories for each topic in the file system. They are unique within the underlying map and usually quite short.

Within that directory we bundle all the content pertinent to that topic. For bulk update or retrieval the whole topic information is made accessible as `.xtm` or `.ctm` text files, or any other supported TM serialization format. Again we use a leading dot to indicate to UNIX tools to normally disregard this file. All subject identifiers of a topic are listed in a file named `~`, borrowing from the TMQL [6] notation. Similarly any subject locators are stored in a file `=`. These files always exist, even when there are no locators or identifiers: it is easier to test for empty files than for their existence.

As several names and occurrences can be attached to a single topic, the design choice is to create one directory for each of these, one named `name`, the other `occurrence`; the latter also exists as `oc` only to reduce typing. Having all names in one directory makes it a trivial operation to find all names.

Each individual name or occurrence is modelled as file, containing the text representation of the value (cf. Fig 1). The name of the file is the topic identifier of the type, `name` and `occurrence` being just two special cases. If either of the names or occurrences are scoped, then the scope (only one single topic as scope is allowed in this scheme) is appended to the file name with `@`, or alternatively using a dot.

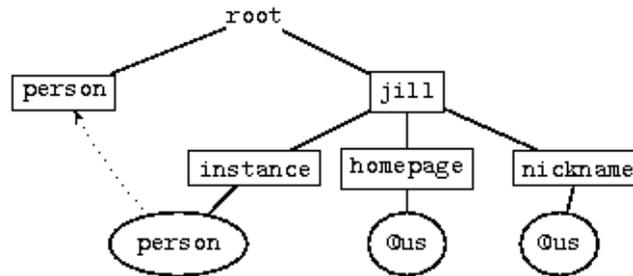


Fig. 1. TMFS structure of a topic

For topics being instances of others, the directory `type` holds symlinks named after the item identifiers of the class topics. For topics used as types, the mechanism is analogous but uses the directory `instance`. For a topic being neither class nor instance, both these directories are empty. Nevertheless these are exposed to an application, because it simplifies later any modification of a topic: a class-instance association can thus be created simply by making a suitable symbolic link in either directory, without having to worry about the directory's existence. As for type-instance configurations between topics, two more directories give access to `superclass` and `subclass` topics.

Regardless whether the topic reifies an association (or a name or occurrence) a file `.reifies` will also exist in the topic directory. If the topic actually does reify any of the above, then `.reifies` will be a symlink to the assertion in question, otherwise the file will be just empty.

3.4 Associations, Names and Occurrences

Associations (but also names and occurrences for that matter) have been tucked away in separate directories, such as `.associations`. In there, each association is itself represented as directory (cf. Fig 2).

In that directory there is always a `.type` symlink to the type of the association, in the same way as there is always a `.scope` symlink to the scope of the association, defaulting to a topic *universal scope* (`us`), if necessary.

As associations can be potentially be reified by a topic, the file system procures a `.reifier` file. Also that always exist, being empty if there is no reification, or being a symlink to the topic in question if there is.

The roles quite naturally are organized into a subdirectory each, with the role topic identifier being used as directory name. Within the role directory are then symlinks, one for each player topic pointing to that topic. Here also lies the one difference between fully-fledged associations and names or occurrences: the latter two also have a file `.equates` which symlinks to the file which represents this characteristic under the appropriate topic.

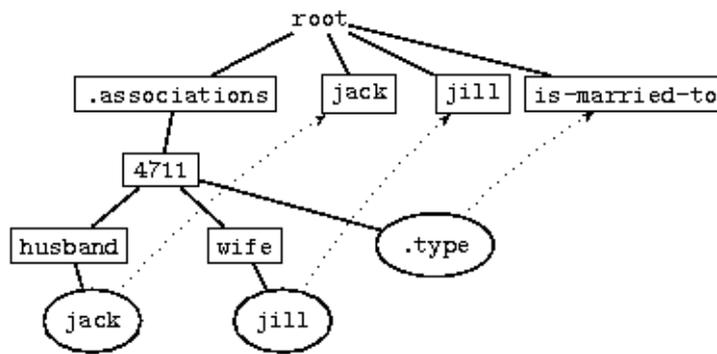


Fig. 2. TMFS structure of an association

4 Canonical Use Cases

In this section we give an overview how Topic Maps, the file system, typical UNIX tools and the compact Topic Map notation together can interact as toolkit to manage Topic Map content at a very small granularity.

4.1 Mounting

In a first step an empty directory has to be created where the topic map structure will reside:

```
mkdir /home/jack/knowledge
```

That directory will act as *mount point* for the topic map content. Everything below that point will be effectively stored in a map. The whole subtree can be provided by a map sphere which has been generated before and is stored in some file, say `/tmp/mapsphere`:

```
mount.tmfs file:/tmp/mapsphere /home/jack/knowledge
```

There are several options concerning the operational model. In some cases the underlying topic map should be used exclusively by the file system. In other cases applications may want to modify the map directly, with changes becoming visible in the file system. Other options control whether map is readonly or not. After mounting the TMFS driver will divert application requests into the file system to the underlying topic map.

To detach the map from the file system, it is *unmounted*:

```
umount.tmfs /home/jack/knowledge
```

4.2 Inspection

After a successful mount a user can list all topics inside the top-level map:

```
cd /home/jack/knowledge/  
ls
```

The result list contains directory file names, one for each topic in the map. To find one particular topic based on its internal identifier can be burdened to pattern matching with a regular expressions:

```
ls j*
```

To learn about all names of one particular topic, the user has to find and concatenate all files in the name subdirectory:

```
cat jill/name/*
```

Each name is represented by a single file named after the names' scope, always prefixed by @. If the scope happens to be unconstrained, then `us` (the unconstrained scope) is used. Each file contains exactly those names in a particular scope. If `jill` had a `nickname` as type for the name, then it can be retrieved via

```
cat jill/nickname/*
```

or - when the scope is to be fixed -

```
cat jill/nickname/@us
```

In an analogous way the user can learn about all occurrences via

```
ls jill/occurrence/
```

and can extract all, say, of type `homepage`

```
cat jill/homepage/*
```

UNIX tools like `find` or `grep` can be used to inspect and query the map. To find all topics which have the nickname *chilly jill* `grep` can be used as follows:

```
grep -i "chilly jill" */nickname/*
```

A full-text search for an *interesting concept* over all occurrences can be achieved with some filtering

```
fgrep -l "interesting concept" */occurrence/*
```

To find which topics use the scope `english` for names or occurrences, `find` can be used:

```
find . -name '@english'
```

Also taxonomic information, such as which superclasses, subclasses, types and instances the individual topic has, have been made available as directories:

```
ls joe/type/
ls joe/instance/
ls joe/subclass/
ls joe/superclass/
```

All will return a list of symbolic links leading to all (direct or indirect) types of `joe`, or instances, sub- or superclasses if there were any. If the user prefers to receive the list of types, instance, etc. in one file, then corresponding files exist in the topic directory as well:

```
cat joe/.types
cat joe/.instances
cat joe/.subclasses
cat joe/.superclasses
```

Should a topic have subject locators or identifiers, these can be retrieved via two specially named files. Both contain potentially a list of URIs, or can be empty:

```
cat joe/~
cat joe/=
```

4.3 Bulk Retrieval

In order to simplify bulk extracts from a map, the file system also procures special files which serialize topic map items (topics, names, occurrences and

associations) in one of the serialization syntaxes. To copy all topic information in the underlying map into one CTM document one can write:

```
cat */.ctm > only-topics.ctm
```

To create an XTM document on the fly the user has only to wrap the XTM fragments:

```
(echo "<topicMap>" ; cat */.xtm ; echo "</topicMap>") > only-topics.xtm
```

Also names and occurrences can be serialized. In

```
cat jack/nickname/.ctm
```

all names of this type, regardless of the scope are collected from this file. To find all names of `jack` and to represent them in AsTMa= (file extension `.atm`), the following suffices

```
cat jack/name/.atm
```

In a similar vein, individual associations offer such files in their respective directory. Also the whole map can be serialized. To do this with the top-level map an application has to read a file in the maps directory, such as

```
cat /home/joe/knowledge/.xtm
```

4.4 Modification

To create a topic in the map, an appropriate subdirectory has to be created:

```
mkdir tmfs
```

First that topic is empty. To add a name in the unconstrained scope, the name string can be directed into the appropriate file:

```
echo "Topic Map File System" > tmfs/name@us
```

Easier it is to use one of the available shorthand text notations and do a bulk insert:

```
cat <<EOT > tmfs/.atm
tmfs isa virtual-file-system-technology
@acronym ! TMFS
! Topic Map File System
EOT
```

The fact that the notation forces to name the topic itself can be used to convey more contextual information about the topic, including associations. This side-effect is quite fortunate as it is not possible to create association directories. If that were allowed, associations without type, scope and roles could be created, something which would lead to an invalid underlying topic map instance.

To guarantee the integrity of associations they are created as *whole*:

```
echo "implemented-with (concept: tmfs, toolkit: perl_tm)"
>> .associations/.ctm
```

It is also possible to incrementally modify associations as long as they always have at least one type and one defined scope, even if it is the unconstrained one.

5 Research Prototype

To test the feasibility and usability of our conceptual translation we have implemented a prototype. It is based on an existing Topic Map distribution, Perl TM [3], and FUSE [13], the *File system in Userspace*.

FUSE is an interface layer between a userland application and the UNIX kernel. It allows to create a file system implementation without having to modify any kernel code. Because of this separation between kernel and the file system code, FUSE has become useful for rapid prototyping and for creating *virtual file systems*, for which there exist a significant number.

Under FUSE the file system handling code is run in user space, with the privileges (and restrictions) of a normal user. Whenever the user accesses the virtual file system, the kernel forwards these accesses to the FUSE process. That is usually encapsulated as a user-executable program which contains the custom callback functions linked to the FUSE library.

On startup this program performs any necessary initialization and mount operations and subsequently enters the FUSE main loop. This FUSE loop listens for notifications of file system accesses from the kernel and schedules the appropriate callback function to implement the particular access.

To produce a particular file system in the FUSE framework one has to provide these callback functions that provide the most common file system-related system calls: `getattr`, `getdir`, `read` and `readlink` are required for read-only access. For write support, the extra functions `mkdir`, `rmdir`, `unlink`, `symlink`, `open` and `write` are necessary. All these callbacks must return the same result types as the respective POSIX system calls of the same name.

As long as the process is active, the file system is available at the specified mount point. The external program `fusermount`, provided with the FUSE library, is used to finally unmount a FUSE file system; at that point the main loop of the program terminates followed by any necessary cleanup operations.

6 Design Issues

The whole design process and the refinement based on implementation experiences was dominated how to balance out topic map information onto a file system. Obviously structural units, such as individual associations or topics lend themselves to be represented as directories. If you factor in, though, a concise TM notation such as CTM, then much of the Topic Map content can be carried inside text files as well. Crucial in controlling the granularity is whether UNIX tools can more easily access and analyse text fragments, compared to navigating through the directory structure or testing for the existence of certain files.

6.1 Symlinks

There has been some reluctance to readily make use of symlinks to reflect the graph nature of a topic map instance. The problem with symlinks is that they increase the risk of loops inside applications when they navigate through the file system structure. Applications have to be aware of symlinks, so they either ignore them altogether or deploy some loop detection algorithm. The former is something which POSIX tools do by default, but this may also mean that a user will not get the expected behavior.

Another problem with symlinks is that there are several different paths to one and the same topic map item. This is also something the application must be aware of, for instance, when string-comparing topic identifiers. On the other hand, using symbolic links is incredibly useful as it allows quick access to a referenced item.

6.2 Empty Files vs. No Files

When a topic reifies an assertion, then a `.reifies` symlink leads to that very assertion. But if it does not, there is also a `.reifies` file, albeit an empty one.

The reason for this choice is that programmatically testing a directory for emptiness is not a common operation for the usual command-line tools, whereas testing files for emptiness is.

6.3 Occurrences and Names

For both the decision was to organize them according to the type. It was regarded the most discriminant criterion and the chosen form also allows to honor subclassing so that a `nickname` also appears under the `name` directory.

One consequence of offering a directory for every available occurrence and name type is that the files inside these directories can carry the values, and that each of these files have to be named. For names and occurrences with a scope this is obviously the scope name, but for those which live in the *unconstrained scope* a default name (`us`) has to be chosen arbitrarily.

One downside of this approach is that there is no direct way to navigate to the type, or the scope topic. For that, the file name of the type has to be extracted first.

6.4 Associations

Associations (and all assertions for that matter) can be reached via their internal identifier, but that is unlikely to be a very frequent access path. Instead it is expected that applications will use association types and there the symlinks of the `instance` directory to find all associations of that type. A similar approach could have been chosen for scopes, but this was left for a more evolved version of the mapping.

The path can be followed in reverse, although in an unsymmetric way. Every association directory holds a symlink to its type (and its scope).

One notable observation is that associations are not easily de/constructable in an incremental way. There has to be a minimal structure consisting of the association directory, the type and the scope and at least one role before further roles can be added. Otherwise the association would be incomplete, something which cannot be normally represented with an underlying TM infrastructure.

6.5 Taxonomies

One frequent usage pattern is to retrieve the type(s) or instance(s) of a certain topic. While this could be implicitly done via following the appropriate associations, this functionality is so inherent in Topic Maps that it is exposed quite prominently in the mapping, and actually twice to support different access paths.

The first one treats a type (respectively instance, subclass and superclass) as a single entry living in a symlink underneath the `type` (respectively `instance`, etc.) directory. Alternatively the files `.types` (respectively `.instances`, etc.) render the identifiers of all types of that topic. Notable here is that subclass transitivity is always honored. One consequence of this approach is that there cannot be an occurrence or name type called `type` (or `instance`, etc.).

6.6 Feature Completeness

Relative to TMDM [7] the mapping covers all features, with the exception of variants (they can be better modelled with typed names) and the reification of roles (which is arguably a seldom used feature).

6.7 Scalability

The general approach of representing a topic map as relatively flat virtual file system inherently carries some potential pitfalls. One of them has simply to do with the number of topics or associations which translates into a high number of directories within the map or the `.associations` directory.

This presents a twofold problem: first these long lists have to be built on the fly, something which not only stresses the backend storage, but which also asks for significant amount of content to be forwarded between several layers for this one request. This is wasteful, especially if only a fragment of the topics are eventually relevant. Secondly, command-line tools, respectively the UNIX shell, are confronted with these long lists. Here inherent length limits cut off these lists, rendering the whole mechanism useless.

A distinct disadvantage is the circuitous procedure for executing system calls via multiple interfaces. First there is a call path into the kernel to reach the FUSE module. That hands over the request back into userspace to invoke the custom-made semantics. Only then the actual backend - in our case the TM engine will be queried and traversed, if necessary after consulting the persistent backend.

The results are passed back the whole invocation chain. It will have to be seen whether the benefits of TMFS can outweigh the overhead.

7 Future Work

One deficit of the proposed mapping is the lack to address topics via subject identifiers or addresses. URIs are a foreign concept to file systems.

While our implementation experiences using a Perl-based infrastructure were generally positive, our prototype still has several shortcomings on its own:

- Currently it only allows read-access to the underlying map and offers no control over operational modalities. Write-support will further stress-test the usability of the chosen mapping.
- It also does not completely expose the functionality of hierarchically organized topic maps as defined by the map sphere.
- Support for serialization formats such as AsTMA=, XTM 2.0 and CTM are only rudimentary yet.

Apart from building some smaller applications on top of TMFS we also consider to extend the TMFS functionality in several directions. One direction is to find a way to directly add documents themselves to the file system, not only their references into a virtualized topic map. A command

```
cp tmfs-presentation.pdf /home/joe/knowledge/tmfs-
presentation
```

could trigger TMFS into a behavior to not only create the topic `tmfs-presentation` but also to store the PDF document in some background persistent store. The URL generated there for the new document will be used as subject address for the topic.

While useful by itself, it also opens the venue to overlay topic map content with conventional file content. This can be achieved with a *stacked file system* which allows to put one file system onto another. In our case we would start with a conventional file system to store our documents and would overlay a TMFS to provide meta data for these documents.

References

1. *Portable Operating System Interface (POSIX)—Part 2: Shell and Utilities (Volume 1)*". Information technology—Portable Operating System Interface (POSIX)". pub-IEEE, 1993.
2. International Organization for Standardization, ISO/IEC 13250, Information technology - SGML applications - Topic Maps, 2000.
3. R. Barta. Topic Map Modules for Perl.
<http://search.cpan.org/dist/TM/>.
4. R. Barta. TMIP, a RESTful Topic Maps interaction protocol, 2005. Extreme 2005 Montreal.
5. R. Barta. Knowledge-oriented middleware using Topic Maps. 2007. TMRA 2007, Lecture Notes in Computer Science (LNAI) Vol. 4999, Springer 2008.
6. L. M. Garshol and R. Barta. ISO 18048: TMQL, Topic Maps Query Language, committee draft. <http://kill.devc.at/system/files/tmql.pdf>.
7. L. M. Garshol and G. Moore. ISO 13250-2: Topic Maps - data model, 2008-06-03. <http://www.isotopicmaps.org/sam/sam-model/>.
8. G. Hopmans and L. Heuer. Wd 13250-6: Information technology - topic maps - compact syntax, CTM. 2008. <http://www.isotopicmaps.org/ctm/ctm.html>.
9. R. Jones. GmailFS, Gmail virtual file system.
<http://richard.jones.name/google-hacks/gmailfilesystem/gmailfilesystem-implementation.html>.
10. B. Schandl. SemDAV: A file exchange protocol for the semantic desktop. *Proceedings of the 2nd Semantic Desktop and Social Semantic Collaboration Workshop at the ISWC 2006, Athens, USA, 2006*.
http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-202/SEMDESK2006_0009.pdf.
11. M. Sintek and G. A. Grimnes. RDF2FS - a Unix file system RDF store. *www.semanticscripting.org, SFSW 2008, 2008*.
<http://www.semanticscripting.org/SFSW2008/papers/5.pdf>.
12. M. Szeredi. SSHFS, secure shell filesystem.
<http://fuse.sourceforge.net/sshfs.html>.
13. M. Szeredi. Filesystem in USER space, 2003.
<http://fuse.sf.net/>.

*Topic Maps and
Information Retrieval*

Facet-based Exploratory Search in Topic Maps

Markus Ueberall and Oswald Drobnik

Telematics Group, Institute of Computer Science,
Goethe-University, D-60054 Frankfurt/Main, Germany,
{ueberall, drobnik}@tm.informatik.uni-frankfurt.de

Abstract. In this contribution, we address exploratory search where a user is faced with an information need concerning a domain he lacks specific knowledge. Based on the work of Delbru et al., which introduced metrics to measure the navigational quality of *automatically selected* facets for RDF data, we apply those findings to the semantically richer TMDM and show how exploratory search functionality can be combined with existing approaches.

1 Motivation

Exploratory search addresses information-seeking problems where a user needs to find out something about a domain where he has a general interest, but lacks specific knowledge. Therefore, he will usually submit tentative queries to a search engine and explore the retrieved information, selectively seeking and passively obtaining clues about his next steps [19]. An exploratory interface allows users to find information without a-priori knowledge of the information space.

Especially if the structure of the data is unknown and/or the dataset in question is large, a visual exploration technique like *faceted navigation* is necessary which does not require the formulation of explicit queries but derives them from the user's selections/navigation decisions. It provides the user with immediate results and also avoids "dead ends" by suggesting restriction values to iteratively narrow down the current view of the information space until a satisfying result is obtained (cf. fig. 1).

The underlying *faceted classification system* enables the assignment of multiple classifications—called *facets*—to an object and the flexible ordering of these classifications in multiple ways rather than following a pre-determined,

taxonomic order. A facet is a metadata attribute that should represent a single important characteristic or property of the classified objects. Intuitive facets describe properties that are either temporal (e.g., *date-of-birth*), spatial (e.g., *located-in*), personal (e.g., *author*), material (e.g., *topic*) or energetic (e.g., *action*). Unfortunately, these facets almost always have to be constructed manually using on-hand ontologies and can only be used efficiently on fixed data structures [3].

In the context of heterogeneous, dynamically changing datasets, however, an automated technique to identify facets—i.e., relationships between objects in the information space—is needed in order to immediately accommodate for changes and provide users with updated, context-sensitive classifications.

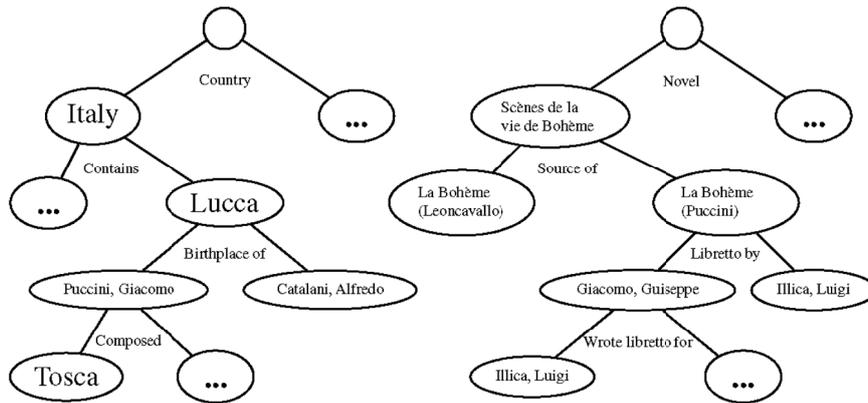


Fig.1. Faceted navigation seen as decision tree traversal: By iteratively choosing a facet and an associated restriction value, the information space is traversed

In the following, we adopt the approach of Delbru et al. [12] which introduces metrics for automatically selected facets for RDF data¹ and show how existing navigation/exploration support found in a number of Topic Map applications can be enriched with exploratory search, shielding the user from the underlying Topic Map based representation of the information space. Section two adapts the original definitions of the metrics with respect to a TMDM representation [10]. The architecture of a TMAPI² based prototype and preliminary experience with its interface is given in section three. Section four discusses the deployment of the forementioned exploratory interface and its perspectives. A summary and an outlook on further work concludes this contribution.

¹ cf. <http://www.browserdf.com>

² Common Topic Map Application Programming Interface, <http://www.tmapi.org/>

2 Facet Selection

A topic map representing the information space can be seen as a graph, the topics being the vertices, and the n-ary associations as well as the occurrences forming the edges. Let $G=(V,E,l_V,l_E)$ be such a graph, where V is the set of vertices, E the set of edges, and l_V and l_E the labeling functions for vertices and edges, respectively.

According to the TMDM [10], all edges are undirected, so there are no designated source and target vertices; however, in order to simplify the following definitions and to emphasise the fact that any navigation implies a direction, G shall—without loss of generality—represent a directed graph where every undirected edge has been replaced by a pair of directed edges pointing in opposite directions. Given an edge, the mappings $source:E \rightarrow V$ and $target:E \rightarrow V$ return the 'source' vertex (i.e., the topic representing the subject from the current point of view) and the 'target' vertex (i.e., another topic referenced by the association in question, representing an object), respectively.

To illustrate these definitions, the well-known 'employment' association example [1,6] is given in LTM notation [7] below. Employer(s) and employee(s) are connected by means of an association, so depending on the chosen *source vertex* for the next exploration step, i.e., either topic `person` or topic `company`, the respective other topic and the reified occurrences connected to the source vertex—`company-website` or `person-job`—are considered the *target vertices*:

```
[employer = "Employer"] [employee = "Employee"]
[employment = "Employment"
 = "Employs" / employer = "Employed by" / employee]
employment([person = "Person"] : employee,
 [company = "Company"] : employer)
[website = "Website"] [location = "Location"] [job = "Job"]
{company, website, "http://company.com/"} ~company-website
{company, website, "http://product.com/"} ~product-website
{company, location, "http://www.frankfurt.de"}
~company-location
{person, location, "http://www.frankfurt.de"}
~person-location
{person, job, [[consultant/programmer]]} ~person-job
```

Contrasting RDF, where a statement is a triple (*subject, predicate, object*) defining the property value (*predicate, object*) for an entity (*subject*) of the information space [3], TMDM offers richer semantics and supports multiple alternative constructs described by quintuples including identity and scope [14, 4]. E.g., instead of representing `job` and `website` by means of occurrences, associations could have been used, too. By effectively treating occurrences as

(binary) associations in the exploratory interface, users are not burdened with representational details.

2.1 Entities, Values, and Facets

Definition 1. An *entity* is a subgraph G' of an information space, extracted by taking all adjacent vertices of a given vertex v , i.e., $G' = (v, V', E', l_V, l_E)$ where $v \in V, V' \subseteq V, E' \subseteq E$ and $\forall e \in E' : source(e) = v \wedge target(e) \in V'$.

Definition 2. A *view* φ is a set of entities of an information space \mathcal{E} .

Delbru et al. [12] use the term *partition* instead of *view*. This might be misleading, since different views need not necessarily be disjoint. Also, the above notion of a *view* is essentially in line with the more extensive view definition we presented in [16].

Definition 3. In a view, a label is associated to one or several edges. A *facet* is a set of labeled edges, i.e., $f_i = \{e \in E \mid l_E(e) = l\}$. F denotes the set of facets of an information space. The *projection facet*: $\varphi \rightarrow F$ returns those facets associated with a view, i.e., $facet(\varphi) = \{f_i \in F \mid \forall e \in \varphi \wedge e \in E, \exists l : l_E(e) = l\}$

Labels can be scoped as demonstrated in the 'employment' example given above in order to reflect the semantics of the respective point of view though incorporating an undirected association, cf. Fig. 2.

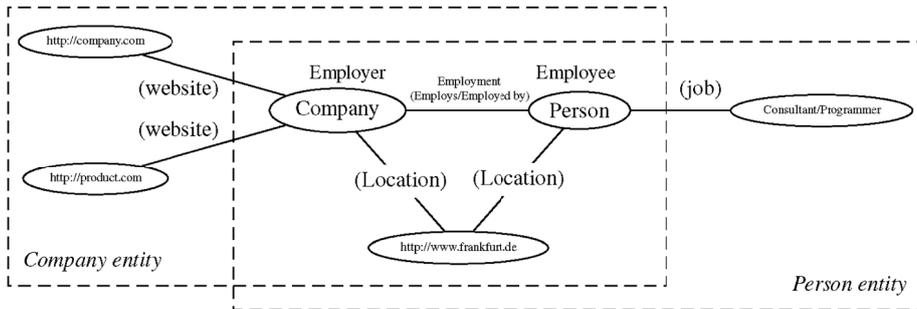


Fig. 2. A view consisting of two overlapping entities, *Company* and *Person*, and associated facets. Unnamed objects are represented by their given topic id in brackets.

Definition 4: The projection $Rv : F \rightarrow V$ returns the set of *restriction values* of a facet, that is, $Rv(f_i) = \{v \in V \mid \exists e \in f_i, target(e) = v\}$. With respect to a set of restriction values of a facet f_i , a view φ can be extracted from the information space. The view implies a new set of facets $F' = facet(\varphi)$, possibly empty.

2.2 Metrics for Navigation

Following Delbru et al. [12], three individual metrics are defined in order to measure the navigational quality of a facet.

Balance: As seen from fig 1, each branching decision optimises the decision power if the tree is well-balanced [20]; the balance of a facet therefore indicates its navigation efficiency. It is computed as the (non-linear) normalised variance of the number $n_s(o_i, f_i)$ of subjects for each object value o_i where $\mu_s(f_i)$ is the vector mean, n_s is the total number of subjects, and $n_o(f_i)$ is the number of different object values for facet f_i :

$$balance(f_i) = 1 - \frac{\frac{1}{n_o(f_i)} \sum_{i=1}^{n_o(f_i)} (n_s(o_i, f_i) - \mu_s(f_i))^2}{1 + \frac{1}{n_o(f_i)} \sum_{i=1}^{n_o(f_i)} (n_s(o_i, f_i) - \mu_s(f_i))^2}, \quad \mu_s(f_i) = \frac{1}{n_o(f_i)} \sum_{i=1}^{n_o(f_i)} n_s(o_i, f_i)$$

Cardinality: A suitable facet has a *limited* amount of object values to choose from. The object cardinality metric is computed as the number of different objects (restriction values) $n_o(f_i)$ for the facet f_i and normalised using a function based on the Gaussian density depending on the parameters μ_o and σ_o :

$$card(f_i) = \begin{cases} 0 & \text{if } n_o(f_i) \leq 1 \\ \exp(-(n_o(f_i) - \mu_o)^2 / 2\sigma_o^2) & \text{otherwise} \end{cases}$$

Frequency: Suitable facets occur frequently inside the collection: the more vertices/distinct concepts (represented by topics, possibly being reifiers) are covered, the more useful the respective facet is in dividing the information space. The frequency is computed as the number of subjects $n_s(f_i) = |\{v \in V | \forall e \in E : l_E(e) = l, source(e) = v\}|$ in the dataset for which the facet has been defined, and is normalised as a fraction of the total number of subjects n_s :

$$freq(f_i) = \frac{n_s(f_i)}{n_s}$$

These metrics can be combined into a final score through (probably weighted) multiplication. As stated in [12], they are solely an indication of usefulness, because they rank facets according to their navigational value, but not according to their descriptive value.

An example of the resulting values for the two entities in fig. 2 is shown in table 1. Evidently, low-ranked facets still have to be displayed in order to cover the entire information space. However, for datasets with a large number of facets, it

is recommended to hide/group them in order to guide the orientation of the users (see below).

facet	balance(f)	card(f)	freq(f)	score
Employment	1.0	0.72615	1.0	0.72615
(website)	1.0	0.72615	0.5	0.36308
(job)	1.0	0.0	0.5	0.0
(location)	1.0	0.0	1.0	0.0

Table 1: Sample metrics for the view consisting of the two entities *Company* and *Person* of fig. 2 ($\mu_o = 10$, $\sigma_o = 10$). The resulting score is the product of all three metrics. (The values will change if additional entities derived from the six vertices are taken into account as well.)

2.3 Additional Facet Classes

As mentioned at the beginning of this section, Topic Map associations differ in compositional granularity from RDF properties. E.g., their expression involves concepts such as role types coupled with role-playing topics [4, 18]. Since a facet browser needs to be able to present the instances of all available types (i.e., topic types, association (role) types, and occurrence types) and the relations between these types need to be made explicit and selectable by the user [9], these additional facet classes have to be considered, too.

The TMDM also defines the concept of scope—albeit it lacks descriptions of the formal semantics [6], which complicates the generic handling as discussed in subsection 3.1. While the introductory employment example merely demonstrated how to use this concept to introduce context-specific labels, its applications (e.g., multilinguality, provenance, opinion, time, audience, filtering) make clear that existing scopes—composed of a set of scoping topics—represent another class of important facets.

All metrics introduced in the previous subsection can be used for these additional classes of facets as well. The only side effect is an overall decrease of the relative navigational value of individual facets as shown in table 1 as their total number increases (cf. role types `employer` and `employee`).

3 Architecture of the Prototype

In order to evaluate the metrics presented in the previous subsection, a TMAPI based prototype has been developed. A text-based open-source version³ is available for interested readers to reproduce the results regarding the exemplary datasets used in this contribution (application of the presented metrics to other Topic Map driven programs is encouraged, the authors appreciate corresponding feedback).

Implementations of query backends are already available, cf. the `tologx` module for the TM4J⁴ engine. Therefore, the processor for the SPARQL query language utilised in the original *browseRDF* prototype could have been replaced by an TM based equivalent. However, we favored a more self-contained solution instead which only uses encapsulated basic TMAPI calls for computing set operations in order to support both the existing and the upcoming revision of that interface. Since all explorative actions are converted into a selection tree, it shouldn't be time-consuming to substitute the forementioned query backends if needed.

The prototype consists of a text-based user interface, a navigation controller (which provides all the functionality required to build a faceted navigation interface), the facet logic (which keeps the current state of the exploration up to date), a facet model (the representation of the facet theory concepts), and an abstraction layer which accesses a TMAPI1 or TMAPI2 compliant topic map engine. The latter also provides hooks for integrating filters and clustering algorithms in order to exclude topic map objects that should not be taken into account and to group both facets and topic map objects that are considered to represent atomic concepts, respectively.

At every iteration, the user may select a facet with or without restriction values (objects) in order to obtain a new view or combine two existing views using `union` and `intersect` operators. It is possible to switch between existing views and display a hierarchical representation of the underlying navigational decisions, which can be modified individually at any time. No (unique) view is ever modified, so it is always possible to backtrack and return to a previous starting point.

³ available for download via <http://www.tm.informatik.uni-frankfurt.de/Plone/Mitarbeiter/ueberall/tm-exploration/>

⁴ Topic Maps 4 Java, <http://tm4j.org>

3.1 Preliminary Experience

In spite of the objective to shield the user from representational details as much as possible and to present a unified navigation interface, it is necessary to provide him with associated information if he wants to. As an example, the different facet classes mentioned in the previous section potentially exhibit varying qualities regarding their “navigational value”, especially when the underlying topics are re-used in different contexts—which in fact should be avoided if possible, cf. [1]. Without using a windowing toolkit, however, the text-based user interface becomes overloaded quickly, therefore it is planned to provide a browser-based interface for the prototype in the run-up to this conference as well.

With respect to the `union` and `intersect` operators, which were introduced in order to conveniently combine results of different exploratory navigation paths, the missing formal semantics of scope are bound to come to the fore as soon as the user selects associated facets: In this case, the new combined view is likely to contain conflicting statements in different scopes or objects that are looked at from multiple perspectives at the same time, which illustrates a prominent example of the “and/or problem” [6]. The only known workaround for this is to provide the user with different selectable “interpretations”, which unfortunately bypasses the objective to shield the user from representational details at least to a certain extent.

Like the forementioned operators, the possibility to modify “sub-selections” (i.e., previous exploratory decisions on a longer navigation path) strictly speaking isn’t supposed to be considered a part of an *exploratory* interface, because these modifications may require additional decisions or lead to empty (intermediate) result sets. However, these options form an important basis of a computer-supported, both representation- and (query) backend-independent formulation of non-trivial queries which can be analysed by the user afterwards. The main benefit in conjunction with an exploratory interface is that the user is immediately able to relate (intermediate) result sets to atomic navigational decisions.

4 Topic Map Exploration

To get a first insight into navigational support, typical generic Topic Map user interfaces as implemented in the Ontopia Omnigator and Vizigator are looked at. Then we demonstrate how a facet-based exploratory interface extension can enrich navigational support. The Italian Opera topic map (revision 1.8) by Steve

Pepper⁵ is well suited as exemplary dataset. Finally, several perspectives are discussed to offer additional improvements.

4.1 Omnigator and Vizigator

The combined interfaces of the Ontopia Omnigator and Vizigator can be regarded typical and rather comprehensive representatives of applications that are able to display the contents of generic topic maps⁶.

While the Omnigator is a general purpose topic map browser which is not recommended for end users but can rather be considered a teaching aid, the Vizigator is designed for graphically browsing and navigating topic maps.

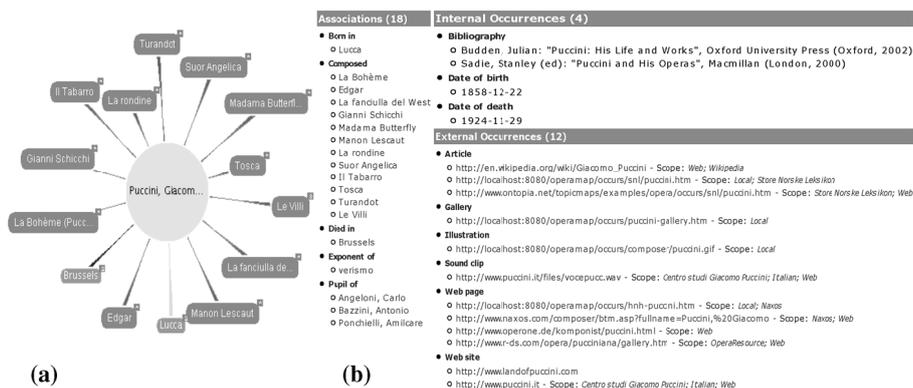


Fig. 3. The vertex representing Giacomo Puccini as shown by the Vizigator (a) and details about associations and occurrences as rendered by the Omnigator (b).

The initial Omnigator view lists all types of a topic map. The text-based browser interface supports navigation between all objects contained in the current topic map, grouping them based on their types according to the TMDM (cf. the lists of associations and occurrences in fig. 3(a)). Known taxonomic information, e.g., existing supertype-subtype relationships, is also displayed. Fulltext searches on names and contents/locators of internal and external occurrences are available.

The Vizigator view of a topic as shown in fig. 3(b) is comparable to the diagram of an entity as outlined in fig. 2. However, occurrences are not handled like

⁵ cf. http://www.ontopedia.net/omnigator/models/topicmap_nontopoly.jsp?tm=ItalianOpera.ltm

⁶ cf. <http://www.ontopia.net/download/freedownload.html>

binary associations, but are listed in a context menu resembling the Omnigator display. Here, only topic names can be searched.

While both interfaces support basic searches, more complex queries which include properties of/relations between topic map objects require the additional use of the tolog query language and thus a certain amount of knowledge regarding the underlying TM representation.

4.2 Faceted Navigation

Using the metrics defined in subsection 2.2, it is possible to group both associations and occurrences into a set of facets. The visualisation of the graph can be restricted to objects that are best suited to support the next exploration step. In this way, the user's orientation in the information space is facilitated. With high probability, dead ends of the search process are avoided. As shown in fig. 4, a view with a single vertex representing composer Giacomo Puccini would only contain five facets representing different "properties". For each set of properties, a limited number of available objects, i.e., possible restriction values is shown. For additional information, the user could either have a look at the individual topic map objects by reverting to a Omnigator/Vizigator like display or extend the current view by selecting additional objects/subjects.

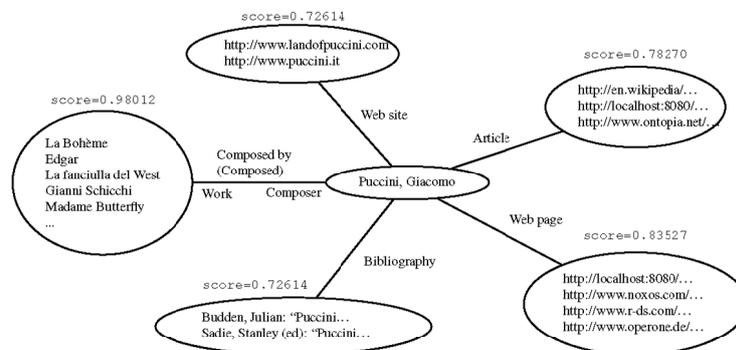


Fig. 4. Stylized display of the vertex representing Giacomo Puccini including only those facets—representing either associations or occurrences—of high navigation quality.

By combining multiple exploration steps or actions which either consist of basic selections of a restriction value, existential selections (i.e., arbitrary values must exist), union, and intersection operations, it is possible to generate a query which is far more powerful than the simple text-based searches. The query can include structural information about the topic map objects while still shielding the user

from the use of a query language and representational details (e.g., associations and occurrences have to be treated differently in tolog). Fig. 5 shows how to determine the name of a certain opera (Tosca) by providing a number of restrictions/constraints.

4.3 Discussion of Perspectives

For large information spaces, the number of facet values may explode. In this case, the navigation process can be improved by reducing the initial number of different facets associated with a view. Two approaches are possible and may be combined for additional user guidance:

1. entities can be partitioned in order to group facets: this requires that, e.g., existing structural knowledge about the domain can be exploited.
2. facet values can be clustered: the clustering has to be computed on the fly and also accommodate for different data types [8].

A good candidate for the first approach is the supertype–subtype relationship as well as a (limited) number of concepts found in the literature that have well-known PSIs [1], which underlines the importance of unified and published processes and concepts for building topic maps [11,17].

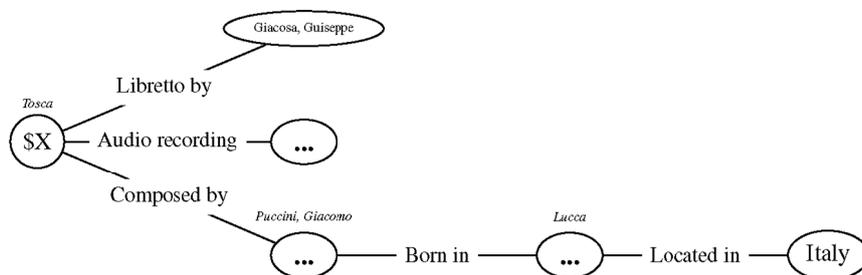


Fig. 5. Example of a resulting complex query generated by combining exploratory actions using basic selection (of a restriction value), existential selection, and join selection with the intersection operator

Aside from the general approach for arbitrary topic maps, an application built around a known ontology like, e.g., the OperaMap Application⁷ is able to filter topic map objects with respect to their usefulness to the exploratory search. Candidates for filtering are metadata or templates which should be presented to the user in a different way. This is especially important for datasets containing versioned concepts [17] which account for a dedicated navigation/user interface

⁷ cf. <http://www.ontopia.net/operamap/>

in order to prevent users from *unintendedly* utilising automatically selected facets that represent explicitly revoked connections.

Additional metrics like the concept of semantic distance between vertices as described in [2] could also be incorporated for determining the (potential) correlation of two automatically selected facets. However, as with filtering, these metrics are likely to be specific to the dataset or application domain. In any case, the three metrics of subsection 2.2 for automatic selection of facets can be used as a fallback.

In order to provide a user with a restricted set of initial topic map objects as a starting point for navigation based on, e.g., a list of interested topics which are not necessarily connected, the creation of a minimal sub-graph—basically a (set of) minimal spanning tree(s) containing the formentioned list of objects—as described in [5] could be considered. Using our current prototype, several manual merge and filter operations are necessary to obtain a comparable view for an initial set of objects which don't share (known) attributes. While their proposed algorithm only operates on existing associations of arbitrary type between topics, the combination of both exploratory interfaces should enable users to faster isolate topic map fragments of interest.

Finally, if the faceted navigation interface were to support the definition and reference of variables acting as sets of restriction values, more complex queries could be generated, e.g., returning a list of all persons within the Italian Opera topic map which were born and died in the same place.

5 Summary and Outlook

The presented exploratory search functionality based on automatically selected facets gives users the chance to inform themselves about an information space without specific domain knowledge. Such kind of interface may enhance existing navigation aids. It is possible to construct queries as depicted in fig. 5 just by following links between concepts of interest, regardless of the underlying query language (e.g., tolog or TMQL) and legend (e.g., the TMDM) which defines how typing information—among other things—is actually represented. As such, this interface can serve as a basis for a user's individual way of looking at concepts and relationships in order to increase the user's productivity as discussed in [13]. Just like the resulting queries, selected navigation paths could be stored in order to provide guidance to users with similar search interests.

Currently, the forementioned functionality is being integrated in the user interface of an Eclipse based prototype for software engineering support, a setting which involves heterogenous user groups with different domain

knowledge [15]. In order to match concepts from different domains, e.g., use cases and design patterns, members of different teams are repeatedly required to trace links between them, thereby crossing the boundary of their own domain of expertise. Exploratory search can complement pre-defined/customisable queries and domain-specific graphical presentation of concepts during the composition and instantiation of templates to come up with a successive formal structuring of information as discussed in [16, 17].

References

1. Ahmed, K.: Topic Map Design Patterns Tutorial. Second International Topic Maps Users Conference (2008).
2. Andres, F., Naito, M.: Application Framework Based on Topic Maps. Proc. 1st International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 3873 (2005) 42–52, http://dx.doi.org/10.1007/11676904_4
3. Delbru, R.: Manipulation and Exploration of Semantic Web Knowledge. Internship Report DERI and EPITA France, July 2006. <http://rdelbru.free.fr/doc/Report.pdf>
4. Dichev, C., Dicheva, D., Dicheva, B., Moran, M.: Translation between RDF and Topic Maps: Divide and Translate. Proc. Balisage: The Markup Conference 2008. <http://www.balisage.net/Proceedings/html/2008/Dichev01/Balisage2008-Dichev01.html>
5. Dichev, C., Dicheva, D., Fischer, J.: Identity: How To Name It, How To Find It. Proc. 16th International. Conference on World Wide Web (WWW) 2007. http://www2007.org/workshops/paper_133.pdf
6. Garshol, L. M.: A Theory of Scope. Proc. 3rd International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 4999 (2008) 74–85, http://dx.doi.org/10.1007/978-3-540-70874-2_9
7. Garshol, L. M.: The Linear Topic Map Notation: Definition and introduction, version 1.3 (rev. 1.23, 2006/06/17), <http://www.ontopia.net/download/ltn.html>
8. Hearst, M.: Clustering versus faceted categories for information exploration. Communications of the ACM Vol 49 No 4 pages 59–61, April 2006. <http://doi.acm.org/10.1145/1121949.1121983>
9. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. Proc. 5th International Semantic Web Conference (ISWC). Springer Lecture Notes in Computer Science (LNCS) 4273 (2006) 272–285, http://dx.doi.org/10.1007/11926078_20

10. ISO/IEC WD 13250-2: Topic Maps - Data Model (TMDM), 2008-06-03, International Organization for Standardization, Geneva, Switzerland.
<http://www.isotopicmaps.org/sam/sam-model/2008-06-03/>
11. Maicher, L.: Autonome Topic Maps, Dissertation, Leipzig 2007 (in german).
http://www.informatik.uni-leipzig.de/~maicher/publications/DISS_LutzMaicher_german.pdf
12. Oren, E., Delbru, R., Decker S.: Extending Faceted Navigation for RDF data. Proc. 5th International Semantic Web Conference (ISWC). Springer Lecture Notes in Computer Science (LNCS) 4273 (2006) 559–572,
http://dx.doi.org/10.1007/11926078_40
13. Park, J., Cheyer A.: Just for Me: Topic Maps and Ontologies. Proc. 1st International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 3873 (2005) 145–159,
http://dx.doi.org/10.1007/11676904_13
14. Schmitz-Esser, W., Sigel, A.: Introducing Terminology-based Ontologies. Proc. 9th International Conference of the International Society for Knowledge Organization (ISKO), 2006. <http://eprints.rclis.org/archive/00006612/>
15. Ueberall, M., Drobnik, O.: Collaborative Software Development and Topic Maps. Proc. 1st International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 3873 (2005) 169–176,
http://dx.doi.org/10.1007/11676904_15
16. Ueberall, M., Drobnik, O.: On Topic Map Templates and Traceability. Proc. 2nd International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 4438 (2006) 8–19,
http://dx.doi.org/10.1007/978-3-540-71945-8_2
17. Ueberall, M., Drobnik, O.: Versioning of Topic Map Templates and Scalability. Proc. 3rd International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 4999 (2008) 128–139,
http://dx.doi.org/10.1007/978-3-540-70874-2_13
18. W3C Working Group Note: A Survey of RDF/Topic Maps Interoperability Proposals (2006). <http://www.w3.org/TR/2006/NOTE-rdftm-survey-20060210>
19. White, R. W., Marchionini, G., Muresan, G.: Evaluating Exploratory Search Systems. Information Processing & Management, Volume 44 Issue 2 (2008) 433–436,
<http://dx.doi.org/10.1016/j.ipm.2007.09.011>
20. Xu, R., Wunsch II, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, Vol. 16 No. 3 (2005) 645–678,
<http://doi.ieeecomputersociety.org/10.1109/TNN.2005.845141>

A Topic Maps-based ontology IR system versus Clustering-based IR System: A Comparative Study in Security Domain

Myongho Yi¹ and Sam Gyun Oh^{2*}

¹ School of Library and Information Studies,
Texas Woman's University, P.O. Box 425438, Denton, TX 76204-5438
topicmap@gmail.com²

Department of Library and Information Science, Sungkyunkwan University, Myongryun-Dong
3-53, Jongro-Gu, Seoul, Republic of Korea
samoh@skku.edu

Abstract. Most clustering methods for information retrieval application do not work efficiently when dealing with complicated data. In this paper, we compare the performance of the Topic Maps-based method with the Clustering-based method. An experimental test was carried out using 20 volunteer to evaluate and compare the performance of the Topic Maps-based Information Retrieval system and Clustering-based Information Retrieval system in security domain. The experimental results show that a Topic Maps-based method provides both better recall/precision and shorter search time/search steps.

Keywords: Clustering, Ontology, Recall, Search Time, Topic Maps

1 Introduction

Many information organization approaches such as taxonomy, thesaurus, classification, and ontology have been attempted to provide effective searching. Among them, clustering and ontology approaches have received much attention. However, there have not been many studies which compare in terms of user performance. Previous studies have been attempted to demonstrate the benefits of clustering or ontology. Therefore, the comparison of each Topic Maps-based and clustering-based approach seemed valuable.

The purpose of this study is to compare the performance of our Topic Maps-based method with the Clustering-based method. In order to measure performance, this study implements a Topic Maps-based Security Information Retrieval (TMIR) system and Clustering-based Security Information Retrieval (CIR) system. Recall/Precision, search steps taken, and search time spent for given tasks are measured.

This paper has been organized as follows: section 1.1 will discuss research questions. Section 2 will discuss related works. Section 3 will introduce security domain. Section 4 will describe the development of TMIR and CIR. Section 5 will discuss research design. Section 6 will present the test results. Section 7 will conclude the paper.

2 Research Questions

The purpose of this study is to determine how the Topic Maps-based information retrieval system differs from clustering-based information retrieval system. The study poses the following research questions.

- 1 Are there recall/precision differences between TMIR and CIR?
- 2 Are there search time differences between TMIR and CIR?
- 3 Are there search steps differences between TMIR and CIR?

3 Related Works

Clustering is the classification of data into different subtopic categories. Clustering shows related items according to their similarity. Numerous clustering algorithms have been studied (E.K.F. Dang, Luk, Ho, Chan, & Lee, 2008; Nosovskiy, Liu, & Sourina, 2008). Two main approaches for clustering methods that have been used for data clustering in information retrieval are partitioning and hierarchical (E.K.F. Dang, et al., 2008). One of the limitations of clustering-based approach is that the relationships between terms are still implicit and require prior knowledge to make a relevance judgment. In other words, clustering-based search engines provide related terms by various algorithms; it shows gaps between clustered and user's categories.

Cluster-based search engines differ from ontology. While ontology explicitly reveals equivalence, hierarchical, and associative relationships to the user, cluster-based search engines only show related terms. For example, a user's

search for “security” using a cluster-based search engine retrieves “homeland security,” “security services,” “security resources,” etc. Ontology shows various relationships including related terms; therefore, the user’s judgment about relevance is better supported. One way to minimize the gap between system and user is to add rich semantic relationships among terms.

While clustering attempts semantic clustering, there is an absence of evidence about which semantic clustering can enhance searching. While many researchers address the potential of clustering (Biren Shah, Raghavan, Dhatric, & Zhao, 2006; Dunlavy, O’Leary, Conroy, & Schlesinger, 2007; Hu, Zhou, Guan, & Hu, 2008; Lin, Li, Chen, & Liu, 2007; Liz Price & Thelwall, 2005; Na, Kang, & Lee, 2007; Niall Rooney, Patterson, Galushka, Dobrynin, & Smirnova, 2008; Oscar Loureiro & Siegelmann, 2005; Ronald N. Kostoff & Block, 2005; Sherry Koshman, Spink, & Jansen, 2006; VicencTorra, Lanau, & Miyamoto, 2006), automatic clustering (Nosovskiy, et al., 2008) , cluster-based data mining (Busygin, Prokopyev, & Pardalos, 2008), and cluster-based information retrieval (Kang, Na, Kim, & Lee, 2007), there are few studies that compare user performance with ontology.

Topic maps are one of the two standards ontology languages. Using Topic Maps, users can browse rich semantic relationships among data. Unlike the clustering-based approach, semantic relationships are explicitly shown to users. The assignment of relationship labels to terms can be done automatically. Many structured resources such as metadata, XML, and database schemes contain information that can be automatically converted to terms, term types, and associations. Topic maps have topics that represent subject or terms (Garshol, 2002). Associations are used for linking among topics. An occurrence has actual resources that linked to topics.

4 Security Domain

For this study, security domain was chosen. Security is a complicated domain. Based on information security certification organization, security can be classified into ten domains (ISC, 2008).

- Access Control Systems and Methodology
- Telecommunications and Network Security
- Application and Systems Development Security
- Cryptography
- Security Management Practices

- Computer Operations Security
- Security Architecture and Models
- Law, Investigation, and Ethics
- Business Continuity Planning and Disaster Recovery Planning
- Physical Security

These ten domains can be classified into three broader categories. The first five domains belong to technical security, the next four domains belong to managerial security and the last domain belongs to physical security.

5 Development of Topic Maps-based Information Retrieval (TMIR) and Clustering-based Information Retrieval (CIR)

Development of TMIR and CIR system involves several steps and Figure 1 shows the processes to develop TMIR and CIR systems.

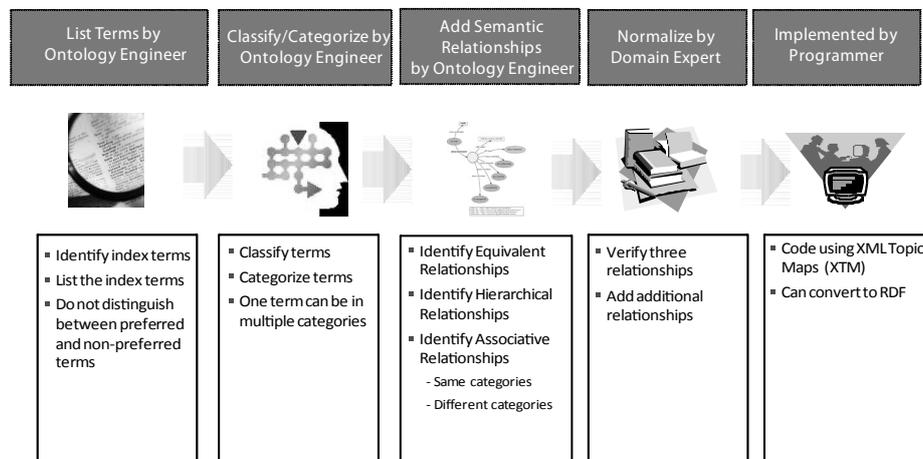


Fig. 1. Development Process of the TMIR and CIR System

5.1 List Terms

The terms are based on the search results from the one of the leading clustering-based search engine, clusty.com (<http://www.clusty.com>). The search with the

term “security” was conducted using clusty.com on June 7, 2008 and the total number of results is 249. The results are classified into two levels (see Table 1). The first level has 48 categories and the second level has 72 categories. The total number of unclassified or so called “Other Topics” is 20.

5.2 Ontology Modeling

Listed terms are converted to Topic Maps-based approach. Table 1 shows two different relationships: clustering-based relationships and Topic Maps-based Relationships. For example, network security (1.1) in clustering-based relationships is classified under information security category and the same network security (1.1) can be labeled as telecommunication, network and Internet security (1.1) in Topic Maps-based approach. The reclassification of clustering-based approach results in well-structured security domain.

Table 1. Data Clustering Ontology

Clustering-based Approach	Topic Maps-based Approach
1. Information Security (16) 1.1. Network Security (3) 1.2. Customers (2) 1.3. Valuable (2) 1.4. PGP (2) 1.5. Other Topics (7) X	1 Equivalent to Security 1.1 Tele., Network & Internet 1.2 Organization/Clients 1.3 1.4 Resources/Standard 1.5
2. Gov (14) 2.1. Social Security Administration (2) 2.2. Department (2) 2.3. Security Police (2) 2.4. Computer Security Resources (2) 2.5. Other Topics (6)	2 Organization 2.1 Organization/Government 2.2 Organization/Government 2.3 Organization/Government 2.4 Resources/Website
3. Alarm (17) 3.1. Monitoring (4)	3 Product/Hardware 3.1 Product/Monitoring

<ul style="list-style-type: none"> 3.2. Security guards (3) 3.3. Equipment, Systems, Surveillance (3) 3.4. Automation (2) 3.5. Intercoms And Access Control Systems (2) 3.6. Focusing, CCTV (2) 3.7. Other Topics (2) 	<ul style="list-style-type: none"> 3.2 Person/Specialty 3.3 Product/Hardware 3.4 Product/Hardware 3.5 Product/Hardware 3.6 Product/Hardware 3.7
<ul style="list-style-type: none"> 4. Bank (15) 4.1. Security Exchange (2) 4.2. Savings Bank (2) 4.3. Security State Bank (2) 4.4. Other Topics (9) 	<ul style="list-style-type: none"> 4 Organization 4.1 4.2 4.3 4.4
<ul style="list-style-type: none"> 5. Homeland Security (8) 5.1. Department of Homeland Security (3) 5.2. Discussion Forums (2) 5.3. Other Topics (3) 	<ul style="list-style-type: none"> 5 Organization 5.1 Organization/Government 5.2 Resources/Dis. Forums 5.3

Ontology modeling is the next step after identifying the data. The ontology modeling process involves building relationships among data. Security ontology modeling is displayed in Figure 2 below.

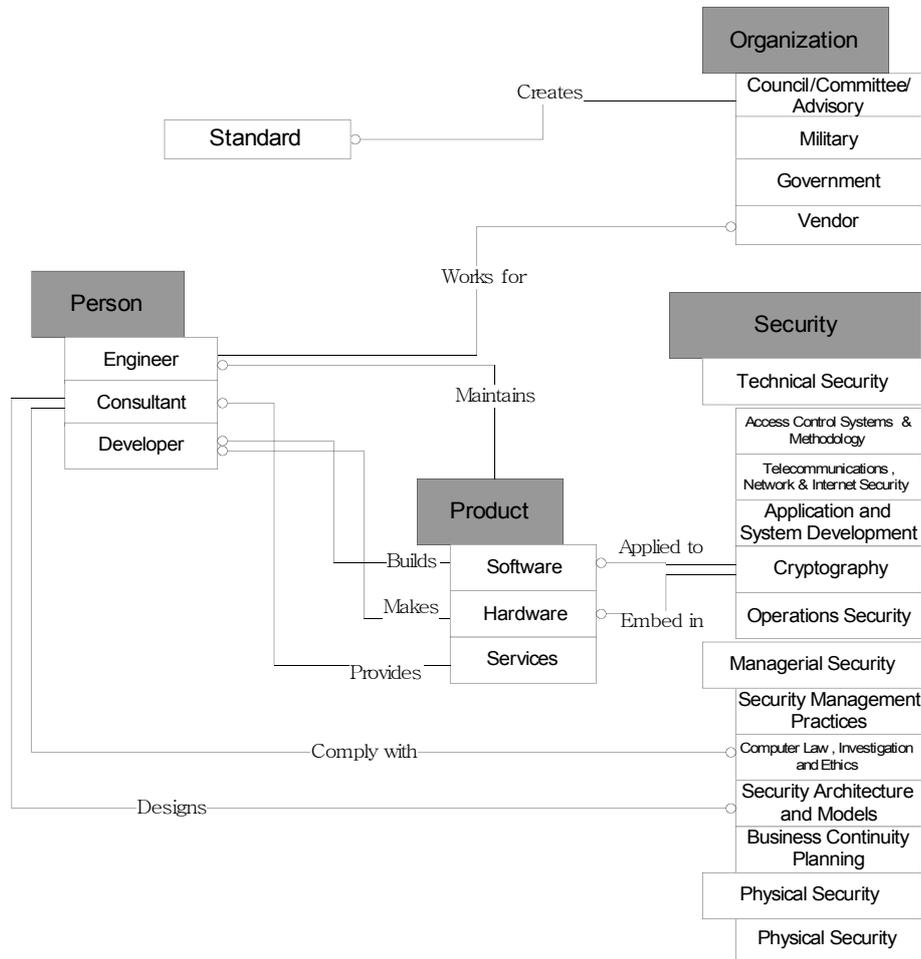


Fig. 2. Security Ontology

Figure 2 shows relationships among terms, and these relationships provide seamless connections among terms. More specifically, when a user searches for software package with specific cryptographic algorithm such as RSA, relationships in Figure 2 allow user to navigate different cryptographic algorithms and related products. The useful association for this search query will be “Applied to” between cryptography and software. Another example for security ontology is as follows: When a user wants to find out an engineer who maintains firewall. A user can find relationships between product and person very easily by browsing relationships between engineer and product.

5.3 Implementation of TMIR and CIR Systems

In order to examine user performance, a TMIR and a CIR system were implemented and a comparative experiment was conducted in which the performance of a TMIR system was compared to that of a CIR system. User performance for both systems was compared and contrasted using an experimental retrieval test, with the only difference between a TMIR and a CIR being a different approach in relationships. Relationships in a TMIR system include equivalence, hierarchical, and two types of associative relationships (both associative relationships between terms belonging to the same hierarchy and associative relationships between terms belonging to different hierarchies). Relationships in a CIR system include various relationships by clustering.

Peter Hancock		Type(s): Engineer
Untyped Names (1)	Internal Occurrences (3)	
<ul style="list-style-type: none"> • Peter Hancock 	<ul style="list-style-type: none"> • Description <ul style="list-style-type: none"> ◦ Peter works for Computer Associates. He is specialized in Intrusion Detection System. He hold CISSP. CISSP Certification was designed to recognize competency in the practice of Information Security. Certification can enhance a professional's career and affirm their level of Information Security mastery and competence. • Email <ul style="list-style-type: none"> ◦ peterh@ca.com • Telephone <ul style="list-style-type: none"> ◦ 1 415 423-1456 	
Associations (2)	External Occurrences (1)	
<ul style="list-style-type: none"> • Maintains <ul style="list-style-type: none"> ◦ Intrusion Detection System • works for <ul style="list-style-type: none"> ◦ Computer Associates 	<ul style="list-style-type: none"> • Homepage <ul style="list-style-type: none"> ◦ http://www.ca.com/peter/ 	

Fig. 3. TMIR

In other words, various associative relationships by clustering exist in CIR system. Based on ontology modeling, a TMIR and a CIR was implemented. In order to implement and navigate ontology, an ontology language and browser were required. Topic maps were used to implement ontology, and “Omnigator” as a topic maps browser was used to demonstrate what we developed for an ontology-driven information retrieval system and a clustering-based information retrieval system. Omnigator is developed by Ontopia and is a free topic map browser that allows users to navigate, test, and debug topic maps (Ontopia, 2005). The name comes from a contraction of “omnivorous navigator.” Omnigator includes a graphic visualization component based on the Vizigator and uses syntax called linear topic map notation (LTM) to build topic maps. Omnigator is based on open standard technologies, in particular XML topic maps

(XTM) and ISO 13250 (Ontopia, 2005). The TMIR and CIR interfaces were designed to be identical and to contain the same domains (Figure 3 and Figure 4).

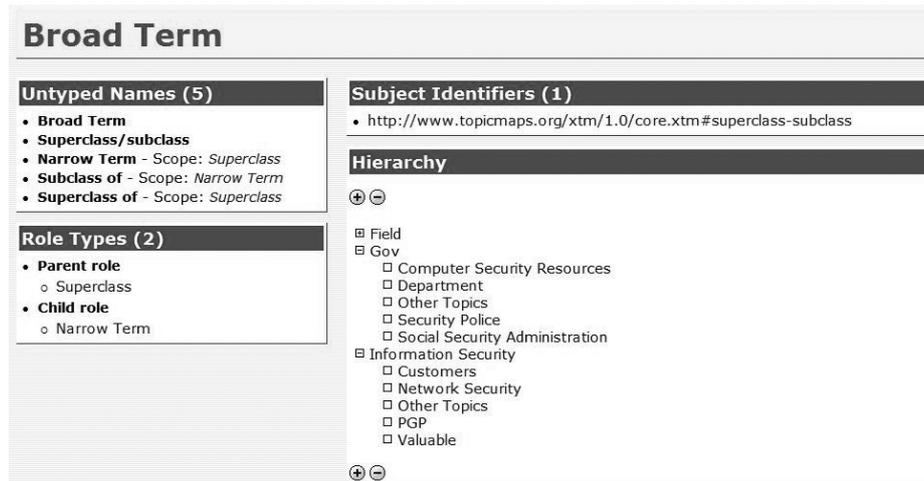


Fig. 4. CIR

6 Research Design

The primary method to address the research questions regarding both systems will be user performance testing. To evaluate the user performance, a comparative experiment will be conducted in which the performance of a TMIR system will be compared alongside a CIR system. A questionnaire about demographic and computer and search engine experiences will be formulated to screen user background. Seven queries will be formulated and distributed to twenty participants to guide their searches using both systems.

6.1 Experiment Participants

Security domain is likely to be used by network or security professionals who deal with various network or security related domains to provide services to clients. Library and Information studies undergraduate students are individuals who may be expected to pursue careers as network or security professionals in the future. Twenty participants (ten for each group) were recruited from the

students who registered for undergraduate courses from multiple higher education institutions for an experimental test.

6.2 Variables

For this study, there was one independent variable: the system (TMIR and CIR). This study included four dependent variables: recall, precision, search time, and search steps. Recall is defined as the percentage of number of relevant documents in relation to the number of relevant documents in the system. Precision is the percentage of relevant documents in relation to the number of documents retrieved. Search time is defined as the period of time devoted to looking for information for the purpose of locating relevant information in response to a request. Search step is defined as the steps to looking for information for the purpose of locating relevant information in response to a request.

6.3 Procedure

Participants conducted searches in a classroom where computers were available. The computers had identical operating systems and browsers. Participants were randomly assigned to either the experimental or the control group. The experimental group was asked to use a TMIR system to search while the control group was asked to use a CIR system. Each group's participants were given the same list of queries and were asked to perform the searches. The tasks included answering queries from the security domain.

The experiment was comprised of four sessions as follows:

1. Pre survey: A questionnaire session about demographic and computer and search engine experiences.
2. A training session including an introduction and a short practice.
3. A test session.
4. Post survey: A questionnaire session about the ease of use, satisfaction and comments

6.4 Data Collection

Two methodologies were used in this study to collect data: Questionnaires and screen recordings. Test sessions are only recorded to analyze recall, precision, search time, and search steps.

6.5 Search Tasks

Search tasks were developed from ontology-modeling. Seven search tasks were formulated and distributed to participants to guide their searches using both systems, as shown in Table 2. These tasks were categorized based on the relationship complexity. The complexity was based on the numbers of concepts, hierarchies, and the degree of relationships between concepts (Byström & Järvelin, 1995). Task categories and task assigned are as follows:

Table 2. Search Tasks

Task #	Degree of Relationships	Task
1	Simple Task	List all the security software
2	Complex Task	Name all the Security engineer who works for Cisco
3	Complex Task	Find Vendors providing security training service
4	Association and Cross Reference Related Task	List all the security hardware supported by IBM Consultants
5	Association and Cross Reference Related Task	List all software using RSA cryptography and find engineer who specializes in these software packages.
6	Association and Cross Reference Related Task	Find security system engineer(s) who specializes in firewall and their supervisor and sale representatives
7	Association and Cross Reference Related Task	Assume that you organization is interested in security training. Who will be the right people to contact? Please provide their e-mail address

To evaluate the TMIR and CIR systems, a comparative pilot study was conducted in which the performance of an Topic maps-based IR system was evaluated alongside a clustering-based IR system in order to determine and then compare

their respective recall, precision, search time, and search steps. The experimental and control groups were given the same tasks and searched for answers using two different information retrieval systems.

7 Results and Discussion

There was a significant difference in recall between the two groups. The estimate value shows the recall on TMIR was higher than CIR. The estimate value also has shown that the search time in the experimental group was less than in the control group. The three research questions were answered with the following conclusions: there were significant differences between the two groups and in terms of recall, precision, search time, and search steps. Overall, recall was higher when performing simple task than when performing complex tasks. The experimental group showed higher recall than the control group. Performing complex-tasks took more search time than performing simple tasks across the two groups. The control group took more total search time than the experimental group.

8 Conclusion

This study illustrates that the positive influences of a Topic map-based ontology IR system are improved recall/precision, shorter search time and search steps for given search tasks than the clustering-based IR system. This study shows that TMIR system resulted in better recall/precision and shorter search times/steps than CIR system. The results of this study attest to the potential of Topic Maps-based ontology to improve information retrieval system performance through better support for associative relationships between terms belonging to different hierarchies by providing explicit relationships among resources.

References

- Biren Shah, Raghavan, V., Dhatri, P., & Zhao, X. (2006). A cluster-based approach for efficient content-based image retrieval using a similarity-preserving space transformation method. *Journal of the American Society for Information Science and Technology*, 57(12), 1694-1707.
- Busygin, S., Prokopyev, O., & Pardalos, P. M. (2008). Biclustering in data mining. *Computers & Operations Research*, 35(9), 2964-2987.
- Byström, K., & Järvelin, K. (1995). Task complexity affects information seeking and use. *Information Processing and Management*, 31(2), 191 - 213
- Dunlavy, D. M., O'Leary, D. P., Conroy, J. M., & Schlesinger, J. D. (2007). QCS: A system for querying, clustering and summarizing documents. *Information Processing & Management*, 43(6), 1588-1605.
- E.K.F. Dang, Luk, R. W. P., Ho, K. S., Chan, S. C. F., & Lee, D. L. (2008). A new measure of clustering effectiveness: Algorithms and experimental studies. *Journal of the American Society for Information Science and Technology*, 59(3), 390-406.
- Garshol, L. M. (2002). What Are Topic Maps Retrieved February 12, 2006, from <http://www.xml.com/pub/a/2002/09/11/topicmaps.html>
- Hu, G., Zhou, S., Guan, J., & Hu, X. (2008). Towards effective document clustering: A constrained K-means based approach. *Information Processing & Management*, 44(4), 1397-1409.
- ISC (2008). CISSP® - Certified Information Systems Security Professional Retrieved June 2, 2008, from <https://www.isc2.org/cgi-bin/content.cgi?category=97>
- Kang, I.-S., Na, S.-H., Kim, J., & Lee, J.-H. (2007). Cluster-based patent retrieval. *Information Processing & Management*, 43(5), 1173-1182.
- Lin, Y., Li, W., Chen, K., & Liu, Y. (2007). A Document Clustering and Ranking System for Exploring MEDLINE Citations. *Journal of the American Medical Informatics Association*, 14(5), 651-661.
- Liz Price, & Thelwall, M. (2005). The clustering power of low frequency words in academic Webs. *Journal of the American Society for Information Science and Technology*, 56(8), 883-888.
- Na, S.-H., Kang, I.-S., & Lee, J.-H. (2007). Adaptive document clustering based on query-based similarity. *Information Processing & Management*, 43(4), 887-901.
- Niall Rooney, Patterson, D., Galushka, M., Dobrynin, V., & Smirnova, E. (2008). An investigation into the stability of contextual document clustering. *Journal of the American Society for Information Science and Technology*, 59(2), 256-266.
- Nosovski, G. V., Liu, D., & Sourina, O. (2008). Automatic clustering and boundary detection algorithm based on adaptive influence function. *Pattern Recognition*, 41(9), 2757-2776.
- Ontopia (2005). The Ontopia Omnigator: User's Guide. Retrieved from <http://www.ontopia.net/download/index.html>

- Oscar Loureiro, & Siegelmann, H. (2005). Introducing an active cluster-based information retrieval paradigm. *Journal of the American Society for Information Science and Technology*, 56(10), 1024-1030.
- Ronald N. Kostoff, & Block, J. A. (2005). Factor matrix text filtering and clustering. *Journal of the American Society for Information Science and Technology*, 56(9), 946-968.
- Sherry Koshman, Spink, A., & Jansen, B. J. (2006). Web searching on the Vivisimo search engine. *Journal of the American Society for Information Science and Technology*, 57(14), 1875-1887.
- VicencTorra, Lanau, S., & Miyamoto, S. (2006). Image clustering for the exploration of video sequences. *Journal of the American Society for Information Science and Technology*, 57(4), 577-584.

Quality, Relevance and Importance in Information Retrieval with Fuzzy Semantic Networks

Roy Lachica¹, Dino Karabeg², and Sasha Rudan³

¹ Bouvet ASA, Norway, roy.lachica@bouvet.no

² Department of Informatics, University of Oslo, Norway, dino.karabeg@ifi.uio.no

³ HeadWare Solutions, Serbia, sasa.rudan@gmail.com

Abstract. We propose a framework for ranking information based on quality, relevance and importance, and argue that a socio-semantic contextual approach that extends topicality can lead to increased value of information retrieval systems. We use Topic Maps to implement our framework, and discuss procedures for calculating the resource ranking. A fuzzy neural network approach is envisioned to complement the process of manual metadata creation.

Keywords: Topic Maps, quality, relevance, importance, semantic search, ontology, resource ranking, information retrieval, contextual search, scoping, tagging, knowledge based systems.

1 Introduction

The Web has enabled an explosive growth of information sharing, but it has also escalated the problem of information overload. The challenge that is now before us is to identify valuable information as judged by the individual user and present the end user with the right information at the right time and place. Organising information by such technologies as Topic Maps answers this challenge only partially, because among the provided topics, associations and resources, some will always be more valuable than others and have different value for different people. In this article we propose a framework for ranking information based on three criteria—quality, relevance and importance—and offer a compound measure called QRI as an extension that can improve the value of information retrieval (IR) systems.

The main objective of IR is the retrieval of relevant information [1]. IR thus becomes a particularly important area for socio-semantic systems, where

perceived irrelevant information has been singled out as a key obstacle to metadata creation [2]. Another related problem is the creation of ontologies which is generally perceived as being time consuming and difficult [3]. There is also the reluctance among both users and institutions to create metadata [2]. We describe how mimicking neural networks to IR can solve these problems.

2 Defining Quality, Relevance and Importance

Our framework refines the conventional view in IR where relevance is the deciding criterion. We employ two additional criteria—quality and importance. In what follows we first survey the ways all three concepts have been treated in literature, and then define them as they are used within the QRI model.

2.1 Quality, Relevance and Importance in the Literature

Based on an analysis carried out by Knight and Burn [4], we identify reliability, availability and relevancy as the main dimensions of quality. According to this study, the quality of information is a compound criterion reflecting a number of specific characteristics.

Table 1. Categories of Information Quality

Reliability	Availability	Relevancy
Accuracy, Concise, Objectivity, Believability, Reputation, Understandability	Security, Accessibility, Navigation, Consistency	Useful, Efficiency, Timeliness, Value-Added, Usability, Amount, Completeness, (Concise)

The notion of relevance is often debated. This concept is both complex and multidimensional. However, in the field of information science, a consensus on the meaning of ‘relevance’ seems to be emerging [1]. Relevance is generally divided into two main categories: topical relevance and user-centred relevance [5]. Topical relevance is objective and mainly concerned with terminology. Topical relevance can be judged by subject area experts. User-centered relevance on the other hand is subjective to the user. Saracevic [6] defines a stratified model of relevance in IR. Relevance occurs on several connected levels. The lower levels concern the interaction with the information system while the upper levels define the user interactions. The upper level consists of: cognitive,

affective, situational and contextual aspects. Situational relevance or utility is the relation between the situation, task, or problem at hand, and the resource. Affective or motivational relevance is relation between the intents, goals, and motivations of a user, and a resource. Cosijn and Ingwersen [7] define sociocognitive relevance as the relation between the resource and the situation, task or problem at hand, as perceived in a sociocultural context. At the top of Saracevic's model is context, which is general and long term. It includes organizational, institutional, community, cultural and historical contexts. Dey [8] uses the term context for any information that can be used to characterize the situation of a user. We use the term context to refer to all the factors that determine what is relevant to a user or group.

Importance as criterion for evaluating information has received little attention in literature. Laudan [9] points at the lack of a viable framework for evaluating this concept, which in part belongs to the realms of philosophy and ethics.

2.2 Quality, Relevance and Importance in the QRI Model

As the above brief analysis shows, quality, relevance and importance have been defined in the literature in a variety of ways. A consequence of this is that the distinctions between those three concepts remain unclear.

Aiming to create a clear-cut set of criteria by which information can be evaluated and ranked by a given user in a given situation, we make the following definition for use in our Topic Maps based framework:

Quality. Quality reflects the intrinsic value of an information resource as judged by an individual. Information that is unreliable or impossible to understand is valueless, even if it may otherwise be highly relevant or important.

Relevance. Relevance is the validity of a relation between two subjects as judged by an individual in a given context. Different persons with different backgrounds might have different opinions about the appropriateness of relations between concepts.

Importance. Importance reflects the degree to which a relation between a user and a subject is valid in a given context. The perceived importance of a subject changes over time as the background and setting of the individual change.

3 Related Research

Research in the crossing point between socio-semantics and contextual information retrieval is scarce. Cantador and Castells [10] propose a multi-layered approach for social applications. Their approach compares user profiles in relation to semantic topics in order to find similarities among users.

Research on ontology based contextual information retrieval is on the other hand more widespread. Context aware relevance ranking can be found in [11], [12]. Stojanovic [13] presents a novel approach for determining relevance in ontology-based search. Siberski [14] discuss why preferences are needed in search and presents a model for use with RDF. Ontology-Based personalisation in IR has been researched by Cantador et al. [15]. Castells et al. [16] also propose the extension of an ontology-based retrieval system with semantic-based personalization techniques. Jrad et al. [17] describe an architecture that provides personalization facilities based on a contextual user model.

4 Assigning Quality, Relevance and Importance

Our model is intended for information retrieval in collaborative knowledge-based systems. Dedicated users create a shared semantic network. All subjects within the system can be used to tag resources. A central feature of the system is listing relevant topics from pages representing different subjects. Ranking of these lists as well as search results lists are seen as the main motivation of the QRI model.

We envision two ways of assigning the value of information w.r.t. each of the criteria: *manual* (by direct input or evaluation) and *automatic* (as a side effect of normal access and use).

We now turn to the central task of how users of the system add the data needed later for resource ranking. The system, for which this framework is intended, should allow users to browse subjects, users and resources. During browsing of the knowledge base the user can assign quality, relevance and importance.

4.1 Manually Assigning QRI

Users can choose to give ratings from 0 to 10 on each of the single QRI criterions. Giving a low rating will hide the topic or association for the user.

Quality. All users can rate the quality of resources. Users will have different rating influence determined by other users through cumulative popular vote. The influence of a user should be a reflection of his trustworthiness, authority, contribution and knowledge level.

Relevance. The user can rate the appropriateness of any subject to subject association. This can be understood as; if the user thinks the association makes any sense. Because relevance is context dependant we add a set of Topic Maps constructs to let the users express their context. Dey and Abowd [18] identify 4 context types:

Location. The location can easily be captured in mobile solutions, but it can also be set in stationary applications by letting the user have a set of popular locations such as ‘at work’ and ‘at home’. The user should have the ability to add locations that are relevant to him in any way. The IP address of the logged-in user might be used to find the location.

Identity. The social and cultural background of the user can be estimated by his or her group membership. The system must therefore support sociability and group management. Identity can also be added through user profile properties like age, education, job, income, etc. although this would have some privacy concerns. The user’s knowledge is also part of his or her identity. This can be supported by having such association types as ‘has knowledge about’ or ‘is expert in’. Pomerol and Brézillon [18] provide an explanation of the relationships between knowledge and context.

Activity. The user cognitive state describes the current situation and mindset of the user. In order to support user cognitive state the system should support various activity related topics such as events, tasks, and projects.

Time. Contextual objects have a start and end date property. Part of the user profile is a local time zone property.

Importance. Users can assign importance to any topic by giving it a rating between 1 and 10. A high rating would imply that the topic is very important to

him. Topics that are important to a user can be listed on the user profile page, on the start page after logging in or similar.

Importance may be set bottom-up or top-down. Bottom-up importance is added by end users. Top-down importance is added by moderators, managers or system owners who want to bias the resource ranking. In some applications it could be desirable to define important information without context. For example a panel of experts might stress awareness to climate change for all members of the system.

4.2 Automatically Assigning QRI

Automatically created QRI may be seen as suggestions made by the system. Automatically created associations receive a relevance weight of one tenth of manually created ones.

Quality. Highly ranked users authoring resources will automatically give a high quality ranking.

Relevance. Simultaneous browsing of two different topics by the same user within a short time span will create a low weight relation. Following an association from one topic to the next will increase relevance.

Importance. Whenever a user browses or use a subject it is marked as important to the user but with a low weight.

5 Topic Map Implementation

There are four master topic types in the ontology: tags, social-items, context-items and resource proxies. Tags are any free subject created by a user. A social item is either a user or a group. Instances of the tag and context item topic type are used as a category or label for tagging resources.

Topic Maps provide an intuitive model for expressing topical relevance. In order to support user-centred relevance we have added the topic types; task, location, event and project along with time span and time zone occurrence types. TMCL is used to constrain allowed associations between topics.

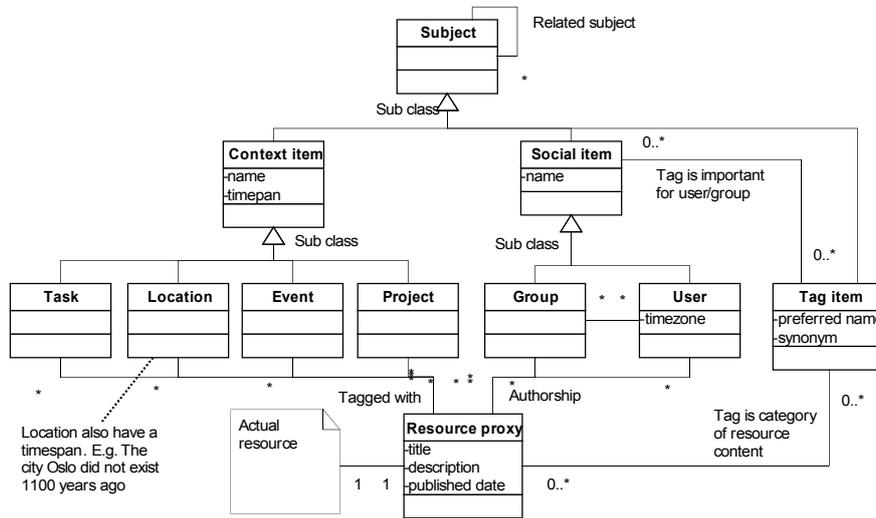


Fig. 1. Basic ontology overview.

5.1 Listing of topics, associations and occurrence types

Table 2. Topic and occurrence types.

Topic type	Occurrence types
Event	Time span
Task	Time span
Project	Time span
Location	Time span, Alias
Tag	Synonyms
Person	Local time zone, Time span
Group	
Resource proxy	

Table 3. Automatically created association types.

Topic type A	Association type	Topic type B
User	Has browsed	[Topic]
User	Has used (tagged, commented, edited)	[Topic]

User	Is browsing from (real world)	Location
User	Has created	[Topic]
User	Has browsed	[Topic]
User	Has communicated with	User
[Topic]	Single user concurrent browsing	[Topic]

Table 4. Manually created association types.

Topic type A	Association type	Topic type B
User	Is to perform *	Task
User	Is friend of	User
User	Is from *, Is currently in/at *, Has been to, Is born in	Location
User	Has authored	[Resource]
User	Has recommended resource	[Resource]
User	Has voted topic as important	[Topic]
Group	Important (favourite)	[Topic]
User	Is to attend *	Event
User	Is member of *	Group Project
User Group	Has knowledge about	[Subject]
[Subject]	Sub-class of, Type of, Is part of	[Subject]
[Subject]	Involves	[Subject]
[Subject]	Is category of (tagging)	[Resource]

[Subject] represents a social-, context- or tag item. [Resource] represents a resource proxy topic which points to an information resource through its subject locator. [Topic] represents any topic what so ever including resource proxies. Transient contextual items (*) switch their association type automatically. If a user create the relation ‘User’ – ‘is to carry out’ – ‘task’, the association will change to ‘has carried out’ when the current local time has passed the time span occurrence value of the item. When choosing to set a topic as important, an ‘important for’-association between the user and the topic is created. Top down importance is created by an expert panel or similar by creating the same relation between a topic and a group.

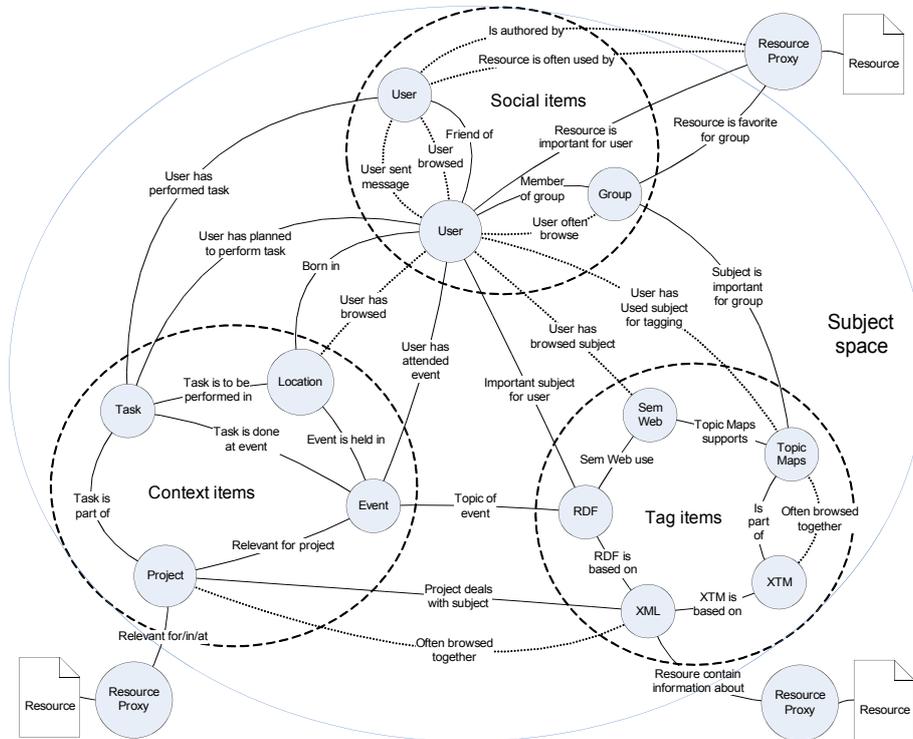


Fig. 2. Sample socio-semantic contextual network. Dotted lines show associations created automatically by the system.

5.2 Mimicking Fuzzy Neural Networks

Knowledge is fuzzy by nature [20]. We use Topic Maps to represent evolving knowledge which is lexically imprecise and/or uncertain. Topic Maps provide a good platform for evolving a knowledge structure similar to that of Collins and Quillian's Semantic Network Model [21]. Central to our neural network approach is Hebbian theory [22], which describes how associations are strengthened with use and weakened when not used. If a user clicks on a tag and he does not find it interesting, it is likely that he will not click on it again and the trail will fade away over time, thus reducing relevance. While most implementations of ontology-based IR rely on bivalent formal engineered ontologies, we utilize a promiscuous approach where users can create semantics ad-hoc. Multiple and overlapping pathways may be created without time

consuming validation or having to adhere to formal schemes that often require high cognitive load on the part of the user. The system learns what is relevant by tracking user interactions and by letting users change the network and its relevance weights. The neural network approach is also used because of the dynamic and complex nature of the user context. Context differs drastically because of surroundings, circumstance, settings, changing goals, the nuances of local and wide global influence. This makes it difficult to have up-to-date information about the context of a user [23]. Our model seeks to solve this problem by automatically evolving a context through the associations growing out from the user topic.

5.3 QRI Implementation

QRI data are kept outside of the topic map data model since it would otherwise demand extreme processing power to process the required context related queries. Also the Topic Maps data model does only allow an association to be scoped by a single topic. Our context tables described below enables a more nuanced scoping by allowing an unlimited number of weighted topics.

Quality. Quality is stored as a rating from 1 to 10 per user giving the rating on resource proxies.

Relevance. Relevance is stored in context tables. These tables will be created for associations if a user decides to rate an association. If for example two parallel associations have been created between the two same topics, the system can decide what is the most relevant for a user based on his context. The context table is populated by retrieving information about the user location, identity, activity, time and knowledge and inserting it into an array. For example a user may be related to several locations through the promiscuous semantic network at the time of defining an important item. The PSI of each location found by CSA (see section 6.2) is added to the location entry along with the semantic distance.

Importance. When an ‘important for’-association is created a context table will be added. This context table will describe in what context the subject is important for the user. This data can then be used for recommending subjects for other users sharing a similar context.

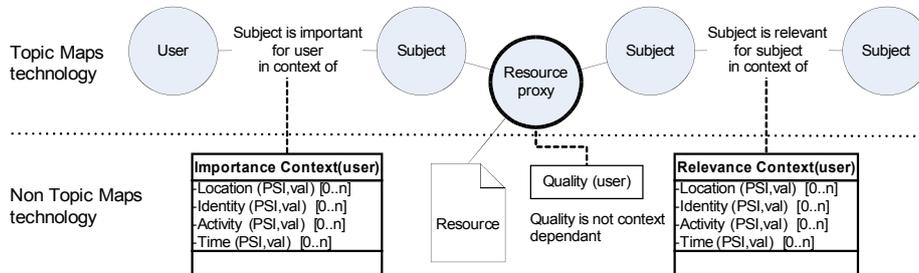


Fig. 3. Conceptual overview of hybrid Topic Map QRI implementation.

The main benefit of this approach is that the current context is captured when creating semantics. This context is then used to give the user information that makes more sense. We envision that this model can also be used to collaboratively evolve ontologies in a bottom up approach. The QRI data can be used in a filtering process to output a consensus topic map.

6 Resource Ranking Calculation

We first describe our general model for resource ranking, and then discuss three scenarios which all have in common the calculation of contextual dependant semantic distances and the use of the quality scores on resource proxy topics. We conclude this section by describing concrete implementations within the fuzzy.com online socio-semantic bookmarking service.

6.1 The Basic Model

Resource ranking is calculated using semantic distance by traversing associations in the topic map. The total semantic distance is measured from the user topic. ‘Important for’-associations act as entry points into the semantic network along side other relations through for example contextual topics.

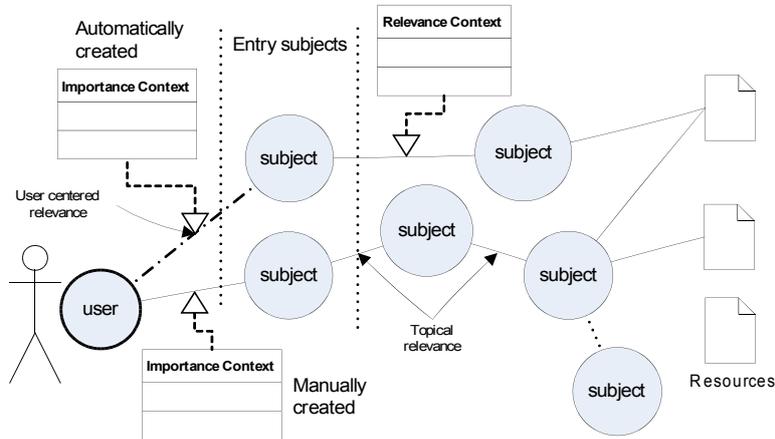


Fig. 4. Simple conceptual overview of semantic distance calculation.

The end result of the ranking will be based on the semantic distance and the quality rating of the resource. Users will have the ability to tune the influence of relevance and quality in the IR process.

Different association types have different weights. When travelling up a ‘class sub-class’ association (more abstract) the weight is decreased more than for other type of associations.

6.2 Ranking in the Context of a Specific Topic

In this scenario, ranking is calculated by following all outward paths from a start-up topic. For each hop, relevance weights are decreased by a configurable factor. All resources above a certain threshold value will be ranked as relevant. The Constrained Spreading Activation (CSA) technique [24] is used for this purpose. A second pass will increase ranking of resources that are related with the user by using the same method of outwards traversal. For contextual topics, relevance weights are automatically adjusted. The ‘Attends’-association shown in figure 3 will have its weight increased when the time of the event is near. Resources found in this process will be ranked by summarizing the relevance score and the quality score assigned to the resource.

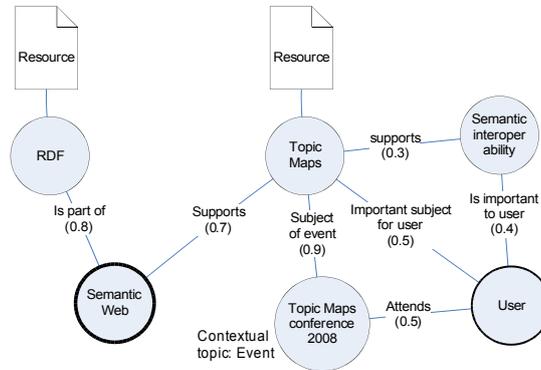


Fig. 5. The starting subject ‘Semantic Web’ has a higher association weight with the ‘RDF’ node in comparison with ‘Topic Maps’. ‘Topic Maps’ will be ranked as more relevant because it is supported by multiple pathways and it is closer in distance to the user making the request.

6.3 Ranking in Keyword Search

Each keyword in the query is matched against topics of the topic map. A syntactic term set enlargement [25] is used to retrieve matching topics by searching preferred names, aliases and using automatic singular/plural nouns. A semantic term set enlargement is performed next using the same spreading activation method as described in the previous section. If a search is performed from a particular subject page, that subject may also be used as an additional start node.

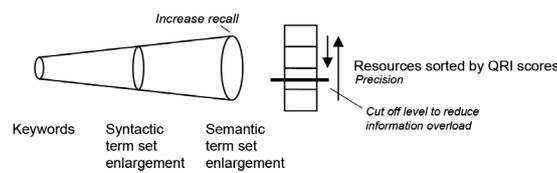


Fig. 6. The keyword search process.

For additional hits a second search may be performed using a keyword match in the resource proxy name and description occurrences.

We will now have a list of subjects that will be used to retrieve relevant resources. The ranking of resources is here calculated using a shortest path algorithm for undirected weighted graphs. The Bellman-Ford algorithm [26] may

be used for this. When there are few hits a third pass should be used for retrieving resources containing the subjects using other indexing search methods.

6.4 Ranking in a Push Scenario

In a push scenario the user is not requesting information. An example of this could be an automated e-mail digest service. The system must use the available context to find what is relevant for the receiver, often referred to as Best Bets systems [27]. In the two above scenarios we can assume the user has already articulated his information needs by browsing or by query formulation. In this scenario, ranking is based on QRI, novelty and user history. The user should only receive lists of new items that he has not already viewed and which are of high importance.

6.5 Feedback loop

Before the topic map is densely populated, ranking in early stages of the system will be inefficient since the required paths between the user topic and the actual relevant resource proxy may not yet exist.

As users visit, use or rank resources, associations between the user and resource proxies are created. Again the Hebbian effect will strengthen relevant associations and less relevant will die out. The relation between the user and the resource will leave a semantic path which will allow other users to find the resource if the user share a similar context.

A timer service or similar mechanism will remove irrelevant information. For each time interval all association weights below a certain level will decrease. The time interval is configurable and should depend on the association/topic ratio.

6.6 Partial Implementation on fuzzy.com

Many of the ideas in this paper have evolved from the issues uncovered in the online bookmarking service fuzzy.com. The current version of fuzzy supports relevance ranking by letting users vote on associations between tags. Users can also define favourite tags, users and resources. This functionality let users directly set items as important to him, but without any context. A resource can also be voted on with a positive or negative vote to indicate quality. Fuzzy has a built in simple contextual semantic search feature. Upon a keyword search, keywords will be matched against all tag names in the system. All tags that have

been created, used, or have been set as a favourite by the user are weighted higher.

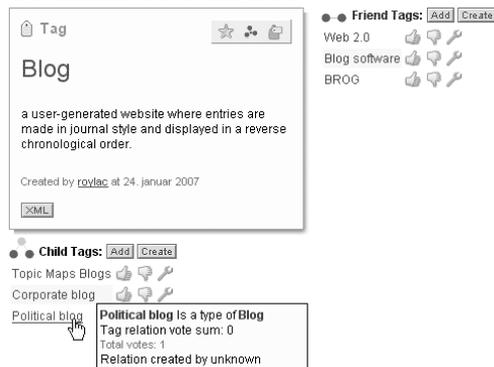


Fig. 7. Voting on associations in fuzzyzy.

Users are able to view the relevance of associated topics as a sorted list and can move associations up or down through voting. Related items below a lower threshold are hidden. Users can create any association they like and it is up to the community to vote for or against the association.

The ideas presented in this paper will gradually be implemented on fuzzyzy.com. Tuning the QRI resource ranking is, among many other areas, a natural continuation of this project along with measuring and benchmarking precision and recall.

7 Concluding Remarks

In this paper we have shown a model for introducing quality, relevance and importance (QRI) in IR with Topic Maps. The model is designed for use in social collaborative systems where concepts such as persons, events, tasks, projects etc. are central. We hypothesize that our neural network approach to IR has the advantage of being intuitive for end-users, as associations can explicitly be shown in the user interface in comparison to other systems where the user does not know why things are listed as relevant. The burden on users to create the underlying semantic network is reduced with a neural network approach where associations are automatically created and evolved both manually and automatically.

Our model introduces a new layer on top of Topic Maps for weighted associations and for advanced contextual scoping which is intended to better

support user context. All these measures together aim to provide the end users with the right information at the right time and place.

References

1. Borlund, P.: The concept of relevance in IR. *Journal of the American Society for Information Science and Technology*, 54(10), (Aug. 2003) 913-925.
2. Lachica, R., Karabeg, D.: Towards holistic knowledge creation and interchange Part I: Socio-semantic collaborative tagging. *Proc. Third International Conference on Topic Maps Research and Application, Leipzig. Lecture Notes in Artificial Intelligence*, Springer: Berlin (2007)
3. Hyvönen, E., Saarela, S., Viljanen, K.: Application of ontology based techniques to view-based semantic search and browsing. In *Proceedings of the First European Semantic Web Symposium, May 10-12, Heraklion, Greece, (2004)*. Springer, Berlin.
4. Knight, S.A., Burn, J.M.: Developing a Framework for Assessing Information Quality on the World Wide Web. *Informing Science Journal*, Vol. 8, (2005) pp. 159-172
5. Kagolovsky Y, Mohr JR.: A new approach to the concept of relevance in information retrieval (IR). In: Patel V, Rogers R and Haux R (editors). *Proceedings of the 10th World Congress on Medical Informatics (Medinfo 2001)*. Amsterdam, The Netherlands: IOS Press, 2001 Sep;10(Pt 1):348-52
6. Saracevic, T.: Relevance: A review of the literature and a framework for thinking on the notion in information science. Part II: nature and manifestations of relevance. *Journal of the American Society for Information Science and Technology*, 58(3), (2007) 1915-1933.
7. Cosijn, E., Ingwersen, P.: Dimensions of relevance. *Information Processing and Management*, 36(4), (2000) 533–550.90.
8. Dey, A.K.: *Understanding and Using Context, Personal and Ubiquitous Computing*, vol. 5, no. 1, 2001, pp. 4-7.
9. Laudan, L.: *Progress and its Problems* (Berkeley, Los Angeles, London: University of California Press, (1971).
10. Cantador, I., Castells, P.: Extracting Multilayered Semantic Communities of Interest from Ontology-based User Profiles: Application to Group Modelling and Hybrid Recommendations. *Computers in Human Behavior, special issue on Advances of Knowledge Management and the Semantic Web for Social Networks*. Elsevier. In press. (2008)
11. Bénédicte Le Grand, Marie-Aude Aufaure and Michel Soto. Semantic and Conceptual Context-Aware Information Retrieval. In the *IEEE/ACM International Conference on Signal-Image Technology & Internet-Based Systems (SITIS'2006)*, Pages 322-332, Hammamet, Tunisie, 17-21 December 2006
12. Aleman-Meza, B., Halaschek, C., Arpinar, I. B., Sheth, A.: Context-Aware Semantic Association Ranking. Paper presented at the *First International Workshop on Semantic Web and Databases*, Berlin, Germany. (2003)

13. Stojanovic, N.: An approach for defining relevance in the ontology-based information retrieval. In: Proceedings of the International Conference on Web Intelligence (WI), Compiègne, France (2005) 359–365
14. Siberski, W., Pan, J.Z., Thaden, U.: Querying the semantic web with preferences. In: Proceedings of the 5th International Semantic Web Conference (ISWC), Athens, GA, USA (2006) 612–624
15. Cantador, I., Fernández, M., Vallet, D., Castells, P., Picault, J., Ribière, M.: A Multi-Purpose Ontology-Based Approach for Personalised Content Filtering and Retrieval. *Advances in Semantic Media Adaptation and Personalization*. Springer-Verlag, *Studies in Computational Intelligence*, vol. 93, pp. 25-51. (2008)
16. Castells, P., Fernández, M., Vallet, D.: An Adaptation of the Vector-Space Model for Ontology-based Information Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19 (2) (2007), pp. 261-272
17. Jrad, Z., Aufaure, M.-A., Hadjouni, M.: A Contextual user model for Web personalization, in: *Personalized Acces to Web Information (PAWI'2007)*, Nancy, france, 3-7 December 2007, 12 p
18. Dey, A., Abowd, G.: Towards a Better Understanding of Context and Context-Awareness, Workshop on the what, who, where, when and how of context-awareness at CHI 2000, April 2000.
19. Pomerol, J., Brézillon, P.: About some relationship between Knowledge and Context. Submitted to the 3rd International Conference on Modeling and Using Context (CONTEXT-01). *Series Lectures in Computer Science*, Springer Verlag. (2001)
20. Zadeh, L.A.: A theory of commonsense knowledge. In H.J. Skala et al., editor, *Aspects of Vagueness*, pages 257–295. Reidel, Dordrecht, 1984.
21. Collins, A.M., Quillian, M.R.: Facilitating retrieval from semantic memory: The effect of repeating part of an inference. In A.F.Sanders (Ed.), *Acta Psychologica 33 Attention and Performance III* (pp. 304-314). (1970) Amsterdam: North-Holland Publ.
22. Hebb, D.O.: *The organization of behavior*, New York: Wiley (1949)
23. Greenberg, S.: Context as a dynamic construct. *Human-Computer Interaction*, 16, (2001), 257-268.
24. Crestani, F., Lee, P.L.: Searching the web by constrained spreading activation. *Information Processing & Management*, 36(4), 2000, 585-605.
25. Kracker, M.: A Fuzzy Concept Network Model and its Applications. In: Proceedings of the FUZZ-IEEE '92, San Diego. pp. 760-768. (1992)
26. Bellman, R.: On a Routing Problem, in *Quarterly of Applied Mathematics*, 16(1), pp.87-90, (1958)
27. Attardi, G., Esuli, A., Simi, M.: Best bets: thousands of queries in search of a client. In *Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers & Posters* (New York, NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM, New York, NY, 422-423.

Standards related research

A case for XTM 3.0

Alexander Mikhailian¹, Rani Pinchuk¹, and Xuân Baldauf²

¹Space Applications Services NV – Belgium
{ami,rp}@spaceapplications.com

²University of Auckland, New Zealand
xuan--x3m3--2008--tmra.de@academia.baldauf.org

Abstract. Improvements to XTM 2.0 are suggested in this paper. First, a set of criteria is defined for evaluating those improvements. It is followed by the suggestions themselves: align element names with the names used in TMDM, reduce the number of elements by introducing mixed content and using attributes whenever it is possible. Finally, some relevant irregularities are discussed.

XTM 2.0[1], The recent standard for Topic Maps exchange is an important step in the direction of popularization of Topic Maps. However, it evolved from the legacy XTM 1.0 [2] format and is lacking clarity at many points. We list below a number of possible changes to XTM 2.0 that have become apparent during the use of XTM 2.0 in our day-to-day work.

The proposed changes to XTM 2.0 will help achieve the following goals:

- 1. Make the format more compact.** XML by itself is quite verbose, so care should be taken not to worsen the situation.
- 2. Improve the parsing speed.** The speed criterion does not need further explanation.
- 3. Simplify the parser development.** By simplification we mean reducing the number of parsing rules.
- 4. Improve the readability.** Although XTM is rather machine-readable, occasional reading of XTM documents by humans for debugging and learning purposes should be possible.
- 5. Improve the learning curve.** Developers should be able to understand XTM easily and quickly, with minimized risk of misconceptions.

We will start incrementally, going from the simple, self-evident improvements to the more critical ones.

Each proposed change will be summarized with regard to the declared goals.

1 Align element names

1.1 rename `itemIdentity`

Element names in XTM 2.0 map unambiguously to those in TMDM. However, there is one element that has a slightly different name in XTM 2.0 than the related term in TMDM. It is `itemIdentity`, which is called *item identifier* in TMDM. It costs nothing to bring it back in line with TMDM.

from XTM 2.0:

```
<itemIdentity href="#shakespeare-wrote-hamlet"/>
```

to XTM 3.0:

```
<itemIdentifier href="#shakespeare-wrote-hamlet"/>
```

There is another case of naming inconsistency. XTM 2.0 uses just name for what TMDM calls *topic name*, but this can be justified, as this element is a child of the `topic` element. The dependence of the name on the `topic` is thus expressed by extralinguistic means.

This change allows to improve on goals 4 and 5.

2 Reduce the number of elements

2.1 introduce mixed content in topic names

In XTM 2.0, the element `name` contains the element `value` which in turn contains the text as `#PCDATA`. The element `value` has no meaning in itself, as it just allows to avoid mixed content. While this made sense at the time when XML processing tools were not mature enough, there is less reason not to use mixed content nowadays, when issues surrounding the mixed content have been widely discussed and understood [4]. We may then remove the `value` element.

Note that the whitespace handling rules for mixed content are not different from those for text content. While editing mixed content by hand, a human editor may be tempted to insert carriage returns and spaces without taking into account the fact that those carriage returns and spaces will be carried on as is by the XML

parser. Fortunately, XTM is not supposed to be directly modified by humans, except for debugging and illustration purposes, as in this paper.

from XTM 2.0

```
<name>
  <value>Shakespeare's authorship of Hamlet</value>
</name>
```

to XTM 3.0

```
<name>Shakespeare's authorship of Hamlet</name>
```

This change allows to improve on goal 2. In general, this change also improves on goal 4, except for the cases where the mixed content is actually mixed, that is, where the content contains type, scope or variant elements. Later in the paper, we will convert the type and the scope elements into attributes, leaving only the variant element. Thus, this change improves on goal 4 except when there are variant elements. Because variants are a rarely used feature in topic maps, we believe that this change is a general improvement on goal 4.

2.2 remove the topicRef element

The `topicRef` element has two slightly different usages. In one usage, it appears as a mandatory child of the `type` element or the `role` element and it may be thought of as a superfluous envelope for the `href` attribute. In the other usage, groups of `topicRef` elements appear as children of `scope` and `instanceOf` elements, each `topicRef` element providing an envelope for the `href` attribute.

In both cases, the parent elements `type`, `role`, `scope` and `instanceOf` indicate the affected property and the `href` attribute determines the value of the property. We may thus drop the `topicRef` element without affecting the data model:

from XTM 2.0

```
<type>
  <topicRef href="#written-by"/>
</type>
...
<scope>
  <topicRef href="#history-of-literature"/>
  <topicRef href="#authorship-issue"/>
</scope>
...
<role>
```

```

    <type>
      <topicRef href="#author"/>
    </type>
    <topicRef href="#shakespeare"/>
  </role>

```

to XTM 3.0

```

<type href="#written-by"/>
...
<scope href="#history-of-literature"/>
<scope href="#authorship-issue"/>
...
<role href="#shakespeare">
  <type href="#author"/>
</role>

```

The element `type` under `role` is mandatory, which allows us to convert it into an attribute. We may also rename the reference to the *association player* from `href` into a more mnemonic `player` attribute:

to XTM 3.0

```

<type href="#written-by"/>
...
<scope href="#history-of-literature"/>
<scope href="#authorship-issue"/>
...
<role player="#shakespeare" type="#author"/>

```

This change is positive for all goals.

2.3 introduce mixed content in variants

The `variant` element can either contain a reference or inline data. This is translated into XTM 2.0 through two elements, `resourceRef` and `resourceData` that can alternatively appear below `variant`. A slightly more compact notation would alter the possible contents of the `variant` element depending on whether we want to use a reference or to paste inline data. The definition of the `variant` element in Relax-NG would then be as follows:

```

variant = element variant {
  (href, reifiable, scope+) | (reifiable, scope+, text)}
data = element data { datatype?, any-markup}

```

And the actual XML would change as follows:

from XTM 2.0

```
<variant>
  <scope>
    <topicRef href="#wikipedia"/>
  </scope>
  <resourceData>Shakespeare authorship question
    </resourceData>
</variant>
<variant>
  <scope>
    <topicRef href="#wikipedia"/>
  </scope>
  <resourceRef
href="http://en.wikipedia.org/wiki/Shakespeare_authorship"/>
</variant>
```

to XTM 3.0

```
<variant>
  <scope href="#wikipedia"/>Shakespeare
  authorship question</variant>
<variant
href="http://en.wikipedia.org/wiki/Shakespeare_authorship">
  <scope href="#wikipedia"/>
</variant>
```

This change allows to improve on goal 2, as well as on goal 4, see section 2.2 for the details.

2.4 introduce mixed content in occurrences

The same reduction of the `resourceRef` and `resourceData` elements can be applied for the `occurrence` element. We will as well convert the `type` element into an attribute.

```
<occurrence type="#wikipedia"
href="http://en.wikipedia.org/wiki/Shakespeare_authorship"/>
```

Just as the previous change, this one allows to improve on goals 2 and 4.

3 Simplify the association

3.1 use attributes whenever possible

We have already started to bring the complex hierarchy of elements under the `association` element to a very compact form by using attributes whenever possible. Let us make the final step and convert the *association type* into an attribute, as well.

from XTM 2.0

```
<association reifier="#shakespeare-wrote-hamlet">
  <type>
    <topicRef href="#written-by"/>
  </type>
  <role>
    <type>
      <topicRef href="#author"/>
    </type>
    <topicRef href="#shakespeare"/>
  </role>
  <type>
    <topicRef href="#work"/>
  </type>
  <topicRef href="#hamlet"/>
</role>
</association>
```

to XTM 3.0

```
<association reifier="#shakespeare-wrote-hamlet"
  type="#written-by">
  <role player="#shakespeare" type="#author"/>
  <role player="#hamlet" type="#work"/>
</association>
```

This change has a major positive effect on all goals.

4 Relevant irregularities

4.1 the `instanceOf` controversy

Until now, we have tried to make the expression of a topic map in an XML document shorter, hoping that a concise representation will bring along readability and will allow for an easier parsing. However, there are cases where the simplification makes for a verbose output.

There is a notorious exception to the way associations are encoded in XTM 2.0. A `type-instance` association can be encoded as a shortcut in the form of an `instanceOf` element. This special case may be unfolded into an `association` element, which would allow to drop the `instanceOf` element from the format.

from XTM 2.0

```
<topic id="shakespeare-wrote-hamlet">
  <instanceOf>
    <topicRef href="#academic-debate"/>
  </instanceOf>
```

to XTM 3.0

```
<association type="#type-instance">
  <role player="#academic-debate" type="#type"/>
  <role player="#shakespeare-wrote-hamlet" type="#instance"/>
</association>
<!--declarations of topics are skipped-->
```

The arguments around the `instanceOf` element are numerous. The summary table below lists several of those:

<i>in favour of instanceOf</i>	<i>against instanceOf</i>
It is by far the most used association type and deserves a special treatment.	It requires implicit knowledge and hardens the learning curve.
Allows for shorter XML and for faster parsing.	Increases the complexity of the parser.
Provides better readability.	Inconsistent with <code>supertype-subtype</code> association type.

The most popular argument in favor of the `instanceOf` element is related to the frequency of its use. The well known Italian Opera [5] topic map contains 1826 `type-instance` associations and only 31 `supertype-subtype` associations. This is a decisive argument. We will *retain* the `instanceOf` element.

from XTM 2.0

```
<topic id="id1">
  <instanceOf>
    <topicRef href="#academic-debate"/>
  </instanceOf>
  <name><value>...</value></name>
</topic>
```

to XTM 3.0

```
<topic id="id1">
  <instanceOf href="#academic-debate"/>
  <name>...</name>
</topic>
```

Abandonment of instanceOf has not been proposed.

4.2 Controversy around `itemIdentity`

The section 3.6 of TMDM [3] states that the `item identifier` is a

...locator assigned to an information item in order to allow it to be referred to.

It has a twofold purpose, and serves as the identifier for the topic map constructs, as well as a way to trace back the origins of the topic map construct, created by merge. This is further explained in the section 5.1 of TMDM [3]:

In a sense item identifiers are identifiers for topic map constructs, but unlike subject locators and identifiers devoid of any specified semantics. Item identifiers may be freely assigned to topic map constructs.

One specific use of item identifiers is in the deserialization from the XML syntax where item identifiers are created that point back to the syntactical constructs that gave rise to the information items in the data model instance.

It is not defined whether the locator is local to the topic map only or universal. However, the reference to the URI [6] and IRI [7] standards for locators in TMDM implies that the universal addressing is at least possible, if not required.

On the other hand, the section 6.2 of [3] explains that during the merging of two topics A and B, a new topic C is created with its `item identifiers` property set

...to the union of the values of A and B's `item identifiers` properties.

This leads to a contradiction that is better explained by the following example of merge of the topics. Let us consider two topic maps, *A* and *B*:

Topic map *A* with the IRI `uri://base1/`

```
<topicMap version="2.0">
  <topic id="id1">
    <subjectIdentifier href="http://www.tmra.de/2008/" />
  </topic>
</topicMap>
```

Topic map *B* with the IRI `uri://base2/`

```
<topicMap version="2.0">
  <topic id="id1">
    <subjectIdentifier href="http://www.tmra.de/2008/" />
  </topic>
</topicMap>
```

Both of these topic maps are merged into a new topic map *C* with the IRI `uri://base3/`.

Topic map *C* with the IRI `uri://base3/`

```
<topicMap version="2.0">
  <topic id="id1">
    <subjectIdentifier href="http://www.tmra.de/2008/" />
    <itemIdentifier href="uri://base1/#id1" />
    <itemIdentifier href="uri://base2/#id1" />
  </topic>
</topicMap>
```

Before merging, there existed exactly one topic with the item identifier `uri://base1/#id1` (the topic in topic map *A*). After merging, however, there exist two topic items with the item identifier `uri://base1/#id1` (the topic in topic map *A* and the topic in topic map *C*). Thus, the item identifier is not universal, anymore. Or, in other words, it can not be addressed from outside of a topic map.

Such a constraint contradicts TMDM [3] in that it effectively enforces a scope on the item identifier which TMDM does not have. It also leaves without any foundation the use of the IRI [7] standard for encoding item identifiers in XTM 2.0 [1].

We solve the contradiction by enforcing the *one topic – one item identifier* principle. We propose that topics have at most one item identifier. When merging two topics *a* and *b* into a new topic *c*, the new topic *c* should get a new item identifier distinct from the item identifiers of *a* and *b*.

Next to the addressing, the second use of item identifiers in XTM 2.0 [1] is to track the origins of a topics. In order to keep this functionality, we introduce a new *item origins* property. This property shall be set to the union of item identifiers of the topics that contributed to the merging. A new topic shall have its *items origins* set empty. A topic created by merging should have its *item origins* property set to the union of item identifiers of the contributing topics.

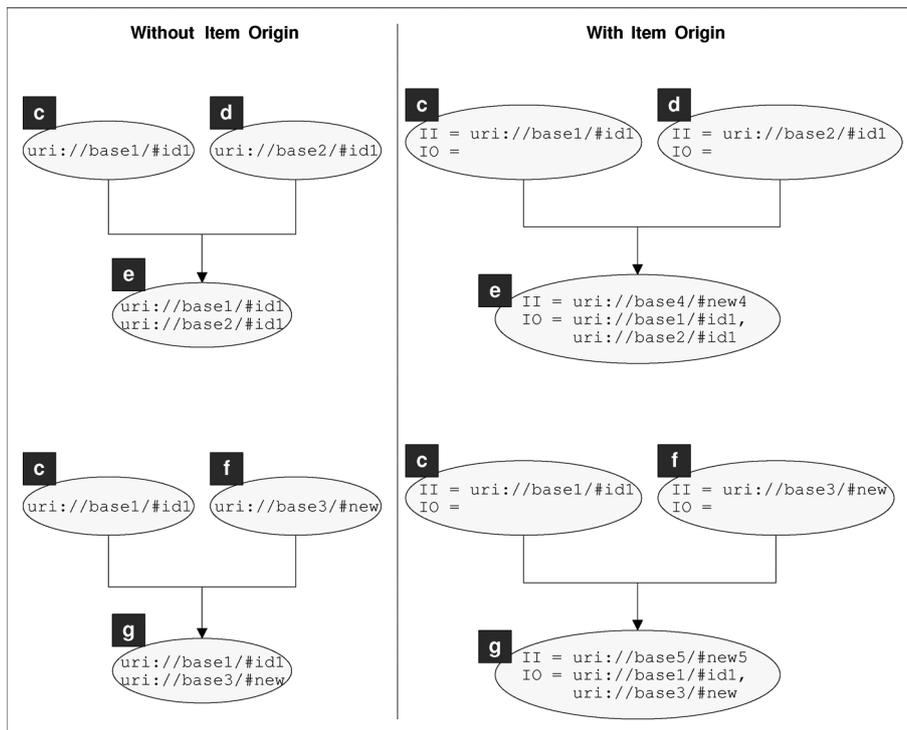


Fig. 1: A more complex use case

The advantage of having *item origins* can be further exemplified by the use case presented in Fig. 1. In this figure, each oval represents a topic from a different topic map. The text inside the topics on the left part represents the *item identifiers* of the topics. On the right side, **II** stands for *item identifier*, and **IO** stands for *item origin*. The arrows between the topics represent merging. For example, the topic **c** is merged with the topic **d** and the result is topic **e**.

As we can see on the left side of the figure, merging **c** and **f** results in the topic **g**. This topic has two item identifiers. One of them is `uri://base1/#idl`. If we try to find the origin of the topic **g** according to this item identifier, we will find that

it can be either the topic *c* or the topic *e*. However, the topic *g* did not originate from the topic *e*.

On the right side, we can clearly identify the origin of the topic *g*, due to the introduction of the item origins.

Implications for XTM 3.0. Because the item identifier of a topic item can be unambiguously determined by the `id` attribute of its `topic` element, we can drop the element `itemIdentifier` altogether. Instead, we introduce the `itemOrigin` element to contain the *item origin* property. The topic map *C* will thus look as follows:

Topic map C

```
<topicMap version="3.0">
  <topic id="id1">
    <subjectIdentifier href="http://www.tmra.de/2008/" />
    <itemOrigin href="uri://base1/#id1" />
    <itemOrigin href="uri://base2/#id1" />
  </topic>
</topicMap>
```

This change positively impacts the goals 3, 4 and 5.

4.3 Ensure completeness

Now that the concepts of *item origin* and *item identifier* have become separate, we are able to use the `xsd:ID` data type for encoding item identifiers and `xsd:IDREF` to point to them. This way, the completeness of the document can automatically be verified at the XML parser level.

Note that using identifiers of type `xsd:ID` has a further advantage. Each `scope` element currently serves just as container for referencing a topic. We replace the list of `scope` elements per statement by a `scope` attribute of that statement. This is possible, because a list of `xsd:IDREF` values, one for each `scope` element, can be represented by one XML attribute of type `xsd:IDREFS`. Such an attribute will contain a list of `IDREF` values separated by spaces. Consider the following example:

from XTM 2.0

```
<topic id="tmra2008">
  <name>
    <scope>
      <topicRef href="#english" />
      <topicRef href="#y2k-pbl" />
    </scope>
  </name>
</topic>
```

```

    </scope>
    <type>
      <topicRef href="#short-name"/>
    </type>
    <value>TMRA'08</value>
  </name>
  <name>
    <value>TMRA 2008</value>
  </name>
</topic>

to XTM 3.0
<topic id="tmra2008">
  <name type="short-name"
    scope="english y2k-pbl">TMRA'08</name>
  <name>TMRA 2008</name>
</topic>

```

This change positively impacts the goals 3, 4 and 5.

5 Conclusion

Not all the goals set at the beginning of the paper can be objectively evaluated. For instance, the readability of the XTM 3.0 documents and a flatter learning curve may only be confirmed by users once the format starts to gain acceptance. The easiness of the parser development shall be evaluated on the actual parser code, coming preferably from multiple implementations.

There is however a way to measure the compactness and, indirectly, the parsing speed by comparing the size of the XTM 3.0 file to the size of the XTM 2.0 file containing the same data. A test on the Italian Opera [5] topic map shows a twofold decrease in the size of the XTM 3.0 document with regard to the XTM 2.0 document.

6 Acknowledgement

This work has been partly funded by the Flemish government through the IWT/ITEA2 project LINDO (ITEA2-06011).

References

- 1 ISO/IEC IS 13250-3:2007: Information Technology - Document Description and Processing Languages - Topic Maps XML Syntax. International Organization for Standardization, Geneva, Switzerland.
<http://www.isotopicmaps.org/sam/sam-xtm/>
- 2 XML Topic Maps (XTM) 1.0 v 1.16 2001/08/06 14:31:44
<http://www.topicmaps.org/xtm/>
- 3 ISO/IEC IS 13250-2:2006: Information Technology – Document Description and Processing Languages – Topic Maps - Data Model. International Organization for Standardization, Geneva, Switzerland.
<http://www.isotopicmaps.org/sam/sam-model/>
- 4 Sean McGrath: Mixed content myopia.
http://www.itworld.com/nl/xml_prac/07112002/
- 5 Italian Opera Topic Map. <http://www.ontopia.net/operamap/>
- 6 RFC 3986, Uniform Resource Identifiers (URI): Generic Syntax, Internet Standards Track Specification, January 2005 <http://www.ietf.org/rfc/rfc3986.txt>
- 7 RFC 3987, Internationalized Resource Identifiers (IRIs), Internet Standards Track Specification, January 2005, <http://www.ietf.org/rfc/rfc3987.txt>

A A sample XTM 3.0 file

```

<topicMap xmlns="http://www.topicmaps.org/xtm/" version="3.0">
  <topic id="shakespeare-wrote-hamlet">
    <subjectIdentifier href="#shakespeare-wrote-hamlet"/>
    <instanceOf ref="academic-debate"/>
    <name scope="wikipedia">Shakespeare's
      authorship of Hamlet<variant>Shakespeare
      authorship question</variant>
    </name>
    <occurrence
      href="http://en.wikipedia.org/wiki/Shakespeare_authorship"
      type="wikipedia"/>
    </topic>
    <association reifier="shakespeare-wrote-hamlet"
      type="written-by"
      id="shakespeare-wrote-hamlet-association">
      <role player="shakespeare" type="author"/>
      <role player="hamlet" type="work"/>
    </association>
    <topic id="wikipedia">
      <name>Wikipedia</name>
    </topic>
    <topic id="written-by">
      <name>Written by</name>
    </topic>
    <topic id="shakespeare">
      <name>William Shakespeare</name>
    </topic>
    <topic id="author">
      <name>Author</name>
    </topic>
    <topic id="hamlet">
      <name>Hamlet</name>
    </topic>
    <topic id="work">
      <name>Work</name>
    </topic>
    <topic id="academic-debate">
      <itemOrigin href="iri://abstract-topics/#academic-debate"/>
      <name>Academic deabate</name>
    </topic>
  </topicMap>

```

B The RelaxNG schema

```

default namespace = "http://www.topicmaps.org/xtm/"
namespace xtm = "http://www.topicmaps.org/xtm/"
datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"

start = topicMap

href = attribute href { xsd:anyURI }
ref = attribute ref { xsd:IDREF }
id = attribute id { xsd:ID }
reifiable = attribute reifier { xsd:IDREF }?, itemOrigin*
datatype = attribute datatype { xsd:anyURI }
version = attribute version { "3.0" }
type = attribute type { xsd:IDREF }
player = attribute player { xsd:IDREF }
scope = attribute scope { xsd:IDREFS }

itemOrigin = element itemOrigin { href }
subjectLocator = element subjectLocator { href }
subjectIdentifier = element subjectIdentifier { href }
instanceOf = element instanceOf { ref }

any-markup =
  (text|element * - xtm:* {attribute * {text}*, any-markup*})*
topicMap = element topicMap
  { version, reifiable, ( topic | association )* }
topic = element topic
  { id, ( itemOrigin | subjectLocator | subjectIdentifier )*,
    instanceOf?, ( topic_name | occurrence )* }
topic_name = element name
  { reifiable, type?, scope?, text, variant* }
variant = element variant
  { (ref, reifiable, scope?) |
    (reifiable, scope?, text) }
data = element data
  { datatype?, any-markup }
occurrence = element occurrence
  { ( href, reifiable, type, scope? ) |
    ( datatype?, reifiable, type, scope?, any-markup ) }
association = element association
  { type, reifiable, scope?, role+ }
role = element role
  { player, type, reifiable }

```


GTM^{alpha} – Towards a Graphical Notation for Topic Maps

Hendrik Thomas¹, Tobias Redmann², Maik Pressler², and Bernd Markscheffel²

¹Knowledge and Data Engineering Group,
School of Computer Science and Statistics, Trinity College Dublin, Ireland
hendriktho@googlemail.com

²Chair of Information and Knowledge Management,
Faculty of Economic Sciences, Ilmenau University of Technology, Germany
{tobias.redmann,bernd.markscheffel}@tu-ilmenau.de
{maik.pressler}@googlemail.com

Abstract. In the last years several drafts, recommendations and concepts for a graphical notation for Topic Maps have been published, but till today no graphical notation is generally approved and used in the Topic Maps community. In this paper we present GTM^{alpha} as a conceptual new notation for a graphical representation of Topic Maps. Our objective is, to provide a practical usable notation, which allows a complete, consistent as well as easy to use graphical representation of any given topic map draft. GTM^{alpha} provides a domain as well as a subject centric view and most important it considers the unique characteristics of the Topic Maps paradigm. This paper serves as a user oriented GTM^{alpha} manual for ontology designers, domain experts as well as users.

1 Introduction

Modeling a Topic Maps ontology is generally a complex and time-consuming process which involves many different actors [1,2]. To support the necessary discussion and to demonstrate modeling options a graphical representation of a topic map draft can be helpful [3, 4]. Using a standardization graphical notation for Topic Maps (GTM) ensures that involved ontology engineers interpret a topic map graphic correctly and uniformly, which is especially important for a collaborative modeling process [5, 6].

In the last years several drafts, recommendations and concepts for a GTM have been published, but till today no graphical notation is general accepted and used in the Topic Map community[3-7]. Recent studies showed that the common trend to reuse and adapt existing graphical notations from the field of data [3]and knowledge modeling [6, 7] is not suitable for this task [8, 9]. Evidences could be identified which indicates that none of these existing notations ¹ could be used for a graphical representation of Topic Maps without adjustments or extensions. However, reusing a existing graphical notation in a different manor as original intended, forces the user to relearn the notation elements and rules as well as significantly increases the risk of misinterpretations. Based on the conducted evaluation we concluded that the creation of a conceptual new graphic notation for Topic Maps is inevitable. A suitable GTM from a pragmatic as well as from a research point of view has to consider general requirements for modeling as well as the specific characteristics of the Topic Maps paradigm [13, 2]. Existing notations can't provide this, because they have been designed for a different purpose and domain.

As a consequence of this insight, we will present in this paper a new conceptual notation draft for the graphical representation of Topic Maps. We designate it as GTM^{alpha} to highlight our objective to provide an every-day usable notation, which allows a complete, consistent as well as easy to use graphical representation of any given topic map draft. This paper serves as an user oriented manual for ontology designers and users who need a graphical representation of a topic map.

In section two we will explain in detail how the GTM^{alpha} should be used and why specific design decisions had to be made to allow a complete and consistent representation of a topic map according to the Topic Maps standard 13250 [2]. In section three we will explain the two pre-defined views of GTM^{alpha}, domain view and the subject centric view. The paper concludes with a summary and an outlook.

2 Manual for GTM^{alpha}

From a scientific point of view a GTM has to allow a complete representation of a topic map. Furthermore the graphical representation should be as consistent as possible to ensure a unambiguous interpretation for users. As a result every graphical representation based on GTM must be transformable into a valid

¹ During the conducted evaluation the notation of concept maps [10], frames [11] and entity relationshippc [12] model have been analyzed in order to determine their suitability to represent a topic map draft under the condition that all individual notation rules are followed by the book.

formal topic map (e.g. XTM, LTM [2]) without losing any information or adding additional elements. A GTM should also provide different views on relevant aspects to simply and support understanding. Considering a cost-benefit-ratio, the amount of effort which is necessary to create a graphical visualization and using the notation should be as low as possible. This has obviously strong impact on the amount of notation elements, the complexity of these elements and modeling rules. Considering the required principle of clearness of a model [8, 14], a graphical Topic Map representation the quality of layout should be high and therefore the GTM design should support the user in the creation of a clear and easy understandable presentation.

Finally, a GTM must reflect the unique characteristics of the Topic Maps paradigm [2, 13]. Essentially two features must be taken into account. First, we have to consider the fundamental rule of Topic Maps: one topic per subject. As a result in a graphical representation a subject should be modeled by exact one element. If this is not possible the notation must provide suitable indicators, which makes clear, that two elements represent the same subject [13, 2]. Furthermore in Topic Maps all kind of types (e.g. topic types, association types, association role types, name types and occurrence types) are represented by topics. As a result a topic can act as a class but at the same time as an instance. This quite unique feature must also be considered in the notation.

Beside these criteria, from a pure pragmatic point of view, a GTM is only suitable for the needs of the Topic Maps community, if it allows to draw a Topic Maps draft fast and easy – with a bad handwriting using a half-full pen on a dirty white board – and an foreign ontology expert is still able to grasp the structure and the elements of the topic map draft correctly and consistently. This is what we need to support the modeling process and communication. Turning to the actual design of a GTM, two observations can be made regarding to these requirements.

First, Topic Maps is clearly a special type of a semantic network consisting of topics and associations representing the relationships between subjects [13]. Consequently the design and main structural principle of the GTM should be network oriented to provide an adequate representation. Second, topics as well as associations are connected to multiple information elements, e.g. a value of a base name, a URI for the subject identity as well as topics acting as association types [2]. From the visualization point of view the resulting network consists of a wide variety of elements, which must be easily identifiable for a user [14].

Coloring the elements might be a good idea. However, typically only a limited number and quite different colors are available, if someone needs to draw something on a black or white board. Even more important is, that a lot of graphical representations of topic maps are made for research publications, like

this one. Those publications are traditionally limited to black and white prints, thus the usage of color is not recommendable for a GTM.

As an alternative we could use, like in previous GTM drafts [5], different node shapes to distinguish Topic Maps constructs. Generally those shapes should be limited to simple geometric forms like ellipses or rectangles to ensure that a topic map draft can be drawn fast and easily. However, the representation of any Topic Maps construct by an individual shape could lead to misinterpretations by the user and to inconsistencies regarding the TMDM[2]. For example, a topic could be symbolized as an ellipse and a scope as a rectangle. This would be suitable to distinguish the elements, but would indicate for a user that the nature of the elements is different.

In fact, both are topics and only one of them acts as a scope in order to define a specific valid context for a Topic Maps construct. Topics can be involved in different roles, e.g. acting as a topic type, scope, association role type, etc. We considered these circumstances in the GTM^{alpha} draft by representing topics with a unique shape and additional symbols indicating which role a topic is playing in the specific construct. Using these approach, a topic is represent by one shape but it can play more than one role, each clearly identified by the specific symbol.

Based on this thoughts the following subsection explains in detail the symbols and notation rules of GTM^{alpha}, which are used to represent the individual topic map constructs. In all further images the topic map elements are labeled. Keep in mind, that these labels are only for education purpose and are not needed in the final graphical visualization of Topic Maps.

2.1 How to represent topics and types relations?

Since the earliest days of Topic Maps [13] a topic has been described as a node in a network and was therefore often graphical represented as an ellipse. In GTM^{alpha} we will continue this good old tradition. Optional a text can be included in the ellipse, which is interpreted as the common name of the topic.² One of the most important constructs in Topic Maps are types, which are used for categorization. In GTM^{alpha} the topic type relation is represented by a arrow line connecting the topic acting as a type and the instance topic. By definition the arrow head points at the topic type element. This arrow line symbol is used in the whole GTM^{alpha} notation to indicate type relationship of all kinds, e.g. the assignment of an association type, role type, name type as well as occurrence types.

² Please note, that the text is actual a shortcut for a base name of the topic without any scope or additional information. More information on this special case can be found in section 2.2. Topic Names.

It is common that a topic type has more than one instances. However, the fundamental rule of Topic Maps demands, that a subject is represented by exact one topic, can make the creation of a suitable layout quite difficult. Because every instance has to be connected with the one topic representing the type in order to be consistent to this rule. As a solution we propose the following approach. Topics acting as types can be drawn more than once, depending on the best position in the graphic for a high quality layout. To clearly indicate for a user, that these multiple topics represent the same subject, inside the type topic a small rectangle containing an ID must be added. All topics representing the same subject must share the same ID. In the formal representation this ID can be preserved as an item identifier, which forces an application to merge the multiple topics automatically [2].

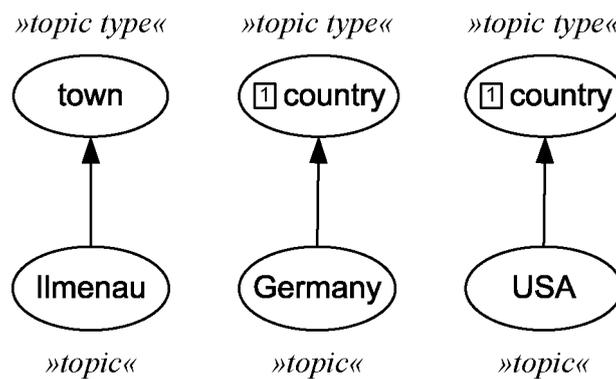


Fig. 1. Topic with topic types

Fig. 1 shows three topics with their according topic types, e.g. Ilmenau is an instance of the class town. As you can see the topic type is represented as topics and only the arrow lines indicates that the topic is used as a type. Germany and USA are both instances of the topic country. The topic type country is drawn twice, but because they share the same ID “1” it is clearly visible for a user that both topic types elements represent the same subject.

2.2 How to represent Topic Names?

As one can see in Fig. 2 the square symbol is used to model topic names. A topic name consists of a base name and optional multiple variant names with specific data values [2, 13]. For a consistent visualization values must be strictly distinguished from topics representing a subjects [8]. Therefore in GTM^{alpha} all

values in a topic map are graphically represented by a rectangle containing the specific value.

A filled square symbol indicates that the value inside a rectangle represents a base name, e.g. the string “Ilmenau” is a suitable name for the topic Ilmenau. Origin from such a base name any number of variant names can be attached, e.g. “IK” and “ILM” as abbreviation for the town name. An empty squares symbol is used to indicate that the value represents a variant name.

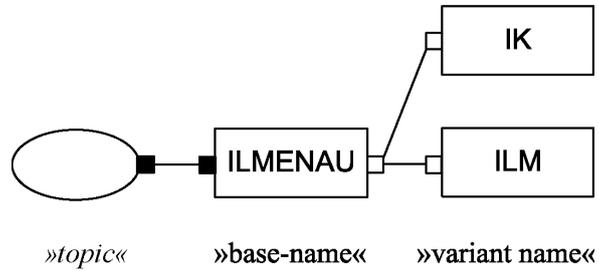


Fig. 2. Base name and variant names

To assign one or more scopes to a topic name, simply attach the topic representing the valid context to the connecting line. A crosshairs was chosen as symbol according to the character of a scope which pin points a valid context. The way to assign a topic name type is similar: simply draw an arrow line to the topic representing the topic type from the connecting line.

Fig. 3 shows how these descriptive elements for a topic name can be modeled. In this example the scope German is assigned to the base name “Ilmenau” as well as the topic name type “official name”.

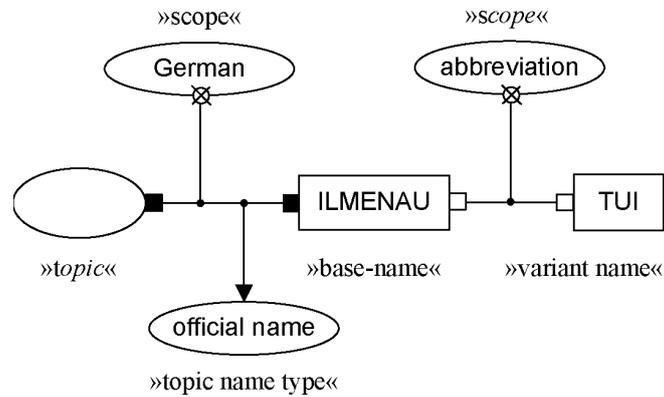


Fig. 3. Topic names with scope, topic name type and data type

In contrast to previous graphics the topic node is drawn empty. This results from a special rule of GTM^{alpha}. The text inside of a topic ellipse node is interpreted as a base name of the topic without any scope or additional information. This is suitable because mostly it is not necessary to draw the complex topic name construct. In many situations it is more helpful to show topics with a standard label. As a results the two graphical representations in Fig. 4. are identical.

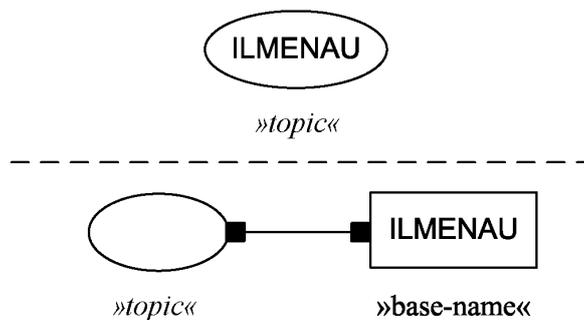


Fig. 4. Topic names shortcut

2.3 How to represent a subject identity?

To symbolize the subject identity of a topic in a graphical representation, we choose a symbol in the shape of the number eight in dependence to the

mathematically symbol for infinity. This should highlight the inescapable bound between a topic and a subject. In addition this symbol is easy to draw and provides an unambiguous indicator. This subject identity symbol can only be used on connections between a topic and a value box containing URI's or in some cases URL's. To indicate that the text inside a value box is a URI, simply underline it. In case of a subject locator, where the URL points to the digital resource the topic is representing, a filled eight symbol is used.

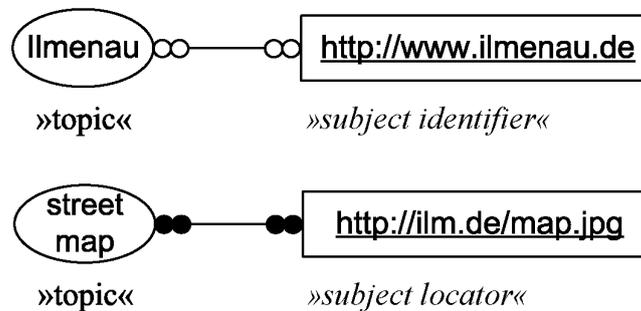


Fig. 5. Subject locator and subject identifier

If a subject identifier needs to be modeled, in terms of that the referred resource acts as an subject indicator, an empty eight symbol is used. This choice was made, because, based on the experiences of the authors, the majority of topics are not direct addressable and therefore subject identifier are more common and an empty eight symbol is a little bit easier to draw than a filled one. The Fig. 5 demonstrates how this notation element should be used, e.g. the URL *http://www.ilmenau.de* is assigned as subject identifier to the Topic Ilmenau and the URL *http://ilm.de/map.jpg* as a subject locator to the topic street map.

2.4 How to represent occurrences?

To represent occurrences in GTM^{alpha} we choose an empty circle as an equivalent to a two dimensional version of the common database symbol. This symbol can be used on connections between a topic and a suitable value element only. An internal occurrence is represented by a rectangle representing the piece of relevant data. For an external occurrence the rectangle must contain a valid URI and must therefore be underlined. The Fig. 6 shows how occurrences are visualized in GTM^{alpha} . In this example the number “98693” is assigned to the

topic Ilmenau as some piece of relevant data. Additionally the web page <http://leipzig.de> is assigned as an external occurrence to the topic Leipzig.

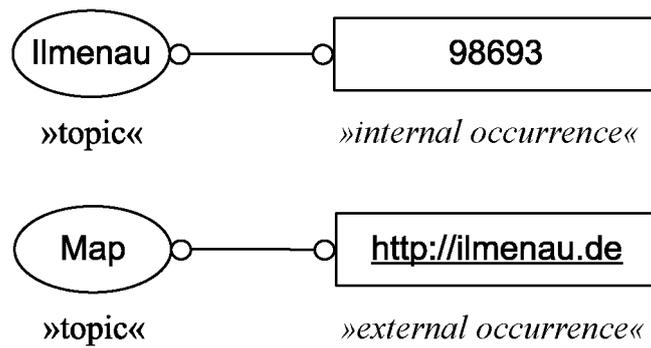


Fig. 6. Internal and external occurrence

Similar to the approach used for topic names additional information can be modeled around the occurrences. Fig. 7 demonstrate how an occurrence type can be assigned by drawing an arrow line to the topic representing the occurrence type, e.g. zip code. Also a scope topic is assigned to the occurrence by using the scope symbol, e.g. limiting the zip code to the context Germany.

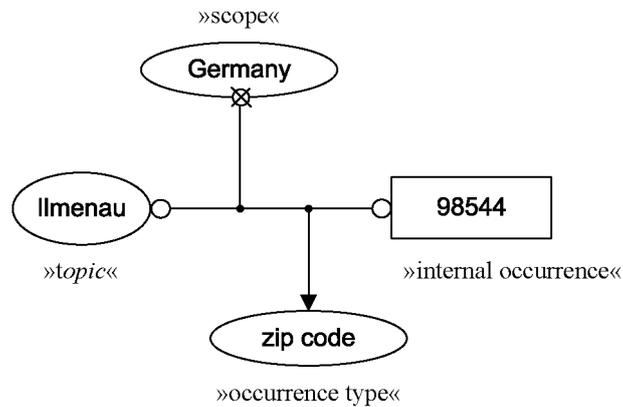


Fig. 7. Occurrence with type and scope

2.5 How to represent an association?

The association is the second fundamental element of Topic Maps, it can be represented in GTM^{alpha} by drawing a line between the associated topics. The additional descriptive elements of the association can be attached directly to the connecting line, which represents the association. In the center of this line an arrow line can point to the association type. The same concept is used to point to the role types for the topic players. The arrow line should originate close to the corresponding topic player. For a better distinction the connecting node for role types should be smaller than the node for the association type. Additionally one or more scope elements can be assigned to the association by drawing a line from center to the scope topic combined with the scope symbol. Fig. 8 demonstrates these recommendations for drawing associations, e.g. the topic Ilmenau plays the role “part” and the topic Thuringia plays the role “whole” in the association “is-part-of”.

During the modeling process an association does not necessarily possess all of these information. Especially in the early modeling stages a modeler often needs only to represent the fact, that there is some kind of association between two topics. In later modeling stages the association is refined and role types as well as the association types are incrementally added. Therefore in the GTM^{alpha} it is allowed to draw an association without role types or without role types and association types.

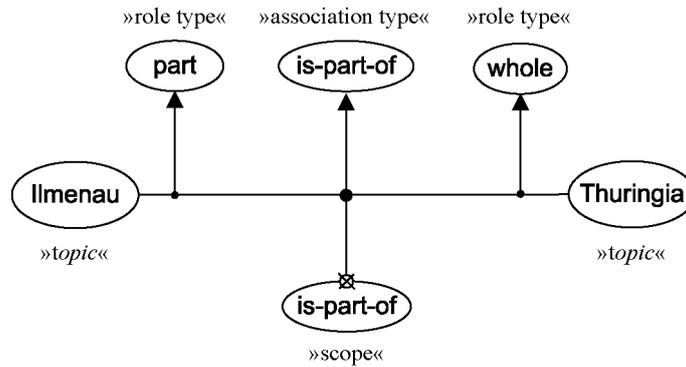


Fig. 8. Association

2.6 How to represent reification?

Sometimes it is necessary to model a statement about other Topic Maps constructs. Such meta-knowledge about the ontology itself can be represented by reification. In terms that a topic represents another topic map construct, like association, a base name, etc. In GTM^{alpha} this can be easily represented by drawing a dotted rectangle around the construct which should be reified. Additionally a dotted line from this rectangle must be drawn to the topic which shall represent the specific construct. As you can see in Fig. 9 the filled eight was chosen as symbol, to indicate that the subject represents the specific topic map construct.

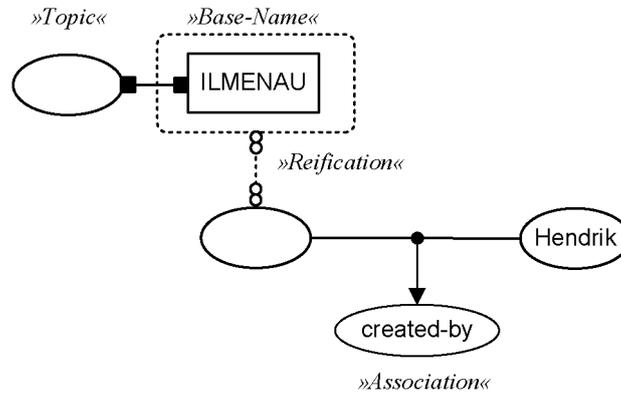
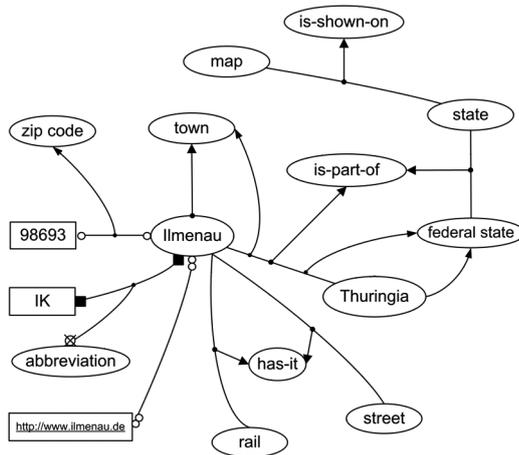


Fig. 9. Reification

3 Domain and Subject Centric Views

The comprehensibility of a graphic model depends on its complexity and its volume. Especially helpful are different views in which only selected aspects of the complex model are visualized. This can reduce the cognitive workload of a user. In the GTM^{alpha} two views are predefined. First, we have the so called *domain view*, which provides an overview of the whole or selected fragments of the modeled domain for a user. A graph oriented structural layout was chosen, in order to highly especially the relationships between the topics in order to allow a user to grasp the big picture. All elements in the domain view are drawn according to the presented GTM^{alpha} notation rules. In Fig. 10 a topic map draft is displayed in LTM and the corresponding GTM^{alpha} domain view.

domain view of the topic map draft in GTM^{alpha}:



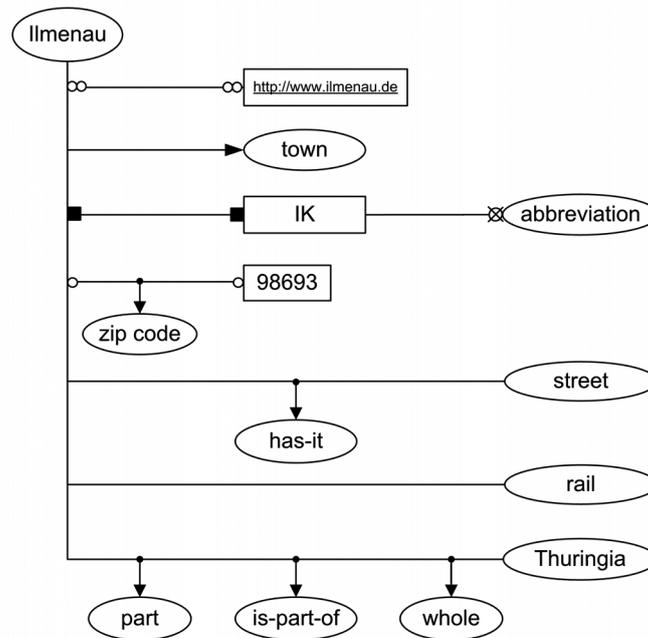
topic map draft in LTM:

```

/*topics*/
[Ilmenau:town = `Ilmenau"; ; /German
 = `IK"/abbreviation)
@`http://www.ilmenau.de"]
[Thuringia: federal state]
[street: traffic connection]
[rail: traffic connection]
[state] [map]
/*associations*/
is-part-of (Ilmenau:town,
Thuringia:federal state)
is-part-of (federal state, state)
is-shown-on (state, map)
has-it (Ilmenau, street)
has-it (Ilmenau, rail)
has-status (Ilmenau, county seat)
/*occurrence*/
{Ilmenau, zip code, [[98693]]}
    
```

Fig. 10. Domain view of GTM

In the topic map developing process very often the discussion is limited to a specific topic rather than to the whole complex topic map network. To provide users a detailed view on the modeled knowledge on a relevant subject in GTM^{alpha} the so-called *subject-centric view* was pre-defined. In this view all topic map elements are drawn according to the presented GTM^{alpha} notation rules, which ensures a cross-model consistent representation and visualization. The layout of the subject-centric view is tree oriented in contrast to the network orientation of the domain view. This enables a well-structured and easy-to-grasp presentation. In the subject centric view the topic of interests is placed at the top and all modeled information are arranged in an explore tree beneath it. We recommend to start with the subject identity, followed by all topic types, topic names, occurrences and at the bottom all association the topic is involved in. Fig. 11 shows the subject-centric view for the central topic Ilmenau of the previous topic map draft. In the end in both views the same information are displayed, only different default layouts are used in order to highlight a specific aspect, e.g. overview or detailed information. Other views are possible.

subject centric view of the topic map draft in GTM^{alpha} :**Fig. 11.** Subject centric view of GTM

The authors have been using the GTM^{alpha} draft successfully for some time. Based on these experiences we identified some best practice recommendations. First the size of the topic element is not pre-defined and can be changed. It is suitable to increase the size of a topic to draw attention to it, e.g. in a discussion or in documentation. The connection line can be straight or curved depending on the layout but crossings of lines should be avoided. The lines for association should be drawn thicker than lines between Topic Maps constructs. Be aware that all symbols and elements can be rotated, without losing their ability to identify an element unambiguously.

4 Summary and Outlook

To support discussion and documentation of the ontology modeling process, we presented in this paper a new draft for a graphical notation for Topic Maps. We showed how GTM^{alpha} should be used to enable a complete and consistent

graphical representation of any given topic map draft according to the TMDM [2]. The amount of effort to learn and use the notation can be considered as moderate. Only few and simple shapes as well as only few notation rule have to be considered.

As the name GTM indicates, the focus was on the design of a practical usable notation. However, it is still a draft and only the broad usage of GTM in the Topic Maps community can lead to a final answer of the question: is GTM really suitable for representing topic map drafts. Overall a GTM has been missing for so long and with this proposal we hope to start a fruitfully discussion, which will finally lead to a official standardized GTM.

References

1. Garshol, L. M.: "Towards a Methodology for Developing Topic Maps Ontologies", in Maicher, L, Siegel, A, Garshol, L. M. (eds.): *Leveraging the Semantics of Topic Maps – Second International Conference on Topic Map Research and Applications, TMRA 2006, Leipzig, Germany, October 11-12, 2006*, Berlin Heidelberg New York, Springer (2007) 20–31
2. Garshol, L. M., Moore, G.: *ISO/IEC JTC1/SC34, Information Technology*, <http://www.isotopicmaps.org/sam/sam-model/> (2006)
3. Gulbrandsen, A. D.: *ORM vs UML for Topic Maps*, <http://www.informatik.uni-leipzig.de/~tmra05/PRES/AG.pdf> (2005)
4. Lee, J.: *Graphical Notation for Topic Maps – Presentation*, <http://www.jtc1sc34.org/repository/0704.pdf>, Seoul, Korea (2005)
5. Henriksen, I.: *Graphical Notation for Topic Maps, Draft 1.2*, <http://cafe.teria.no/ingeh/files/6/13/GTM.pdf> (2006)
6. JTC 1/SC 34/WG 3 ISO/IEC: *N0883 - Draft GTM 13250-7 – Requirements*. <http://www.jtc1sc34.org/repository/0883.htm> (2007)
7. *ISO/IEC ISO 13250-7: GTM (Graphical Notation)* <http://isotopicmaps.org/gtm/> (2008)
8. Becker, J., Rosemann, N., Schütte, M.: "Grundsätze ordnungsgemäßer Modellierung", in *Wirtschaftsinformatik*, 5 (1995) pp. 435–445
9. Schütte, R.: Die neuen Grundsätze ordnungsmäßiger Modellierung: Paper zum Forschungsforum 97, Münster (1997)
10. Noak J. D., Cañas A. J.: *The Theory Underlying Concept Maps and How to Construct Them*, Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition, <http://cmap.ihmc.us/Publications/TheoryUnderlyingConceptMaps.pdf> (2008)

128 Hendrik Thomas, Tobias Redmann, Maik Pressler, and Bernd Markscheffel

11. Reimer, U.: *Einführung in die Wissensrepräsentation: Netzartige und Schema-basierte Repräsentationsformate*, BG Teubner (1991)
12. Chen, P. P.: “The Entity-Relationship Model – Toward a Unified View of Data.” in: *ACM Transactions on Database Systems*, 1 (1), (1976) pp. 9–36
13. Pepper, S.: *The TAO of Topic Maps - Finding the Way in the Age of Infoglut*, <http://www.ontopia.net/topicmaps/materials/tao.html>, Norway (2002)
14. Bennett, W. S.: *Visualizing Software, a Graphical Notation for Analysis, Design and Discussion: A Graphical Notation for Analysis, Synthesis and Discussion*, Marcel Dekker Ltd, Binghampton New York (1992)

TMAPI 2.0

Lars Heuer¹ and Johannes Schmidt²

¹ Semagia
heuer@semagia.com

² INSTANT Communities GmbH
js@sixgroups.com

Abstract. This paper introduces a new generation of the common Topic Maps API (TMAPI) which has evolved from earlier versions based on the Topic Maps Data Model (TMDM) and user experience. TMAPI 2.0 aims to support TMDM and its constraints and to provide a common, user-friendly API for Topic Maps application development independently of a concrete Topic Maps processor.

1 Introduction

Topic Maps API (TMAPI) is a set of Java interfaces and was designed as common programming interface for Topic Maps processors. The initial version was released in the year 2004 and several Open Source and commercial implementations support it. The API was not designed by recognized standards body, but can be seen as a de facto standard for accessing and manipulating topic maps in a portable way. It has been adopted and ported to other programming languages (i.e. PHP5 [5] and .NET [8]) as well.

In the design phase the project members discussed if a programming language neutral approach should be taken for the next TMAPI generation. Even if this idea has its merit it was rejected since each programming language has its own idiomatics and designing an API which meets a common subset of popular languages was felt unpromising. Since the TMAPI project has historically a Java background, the project members opted to focus this language again. Further, the idea that the interfaces should constitute a solid foundation to implement the upcoming standard Topic Maps Query Language (TMQL [3]) on top was also rejected: The project should simply offer an API to access and modify topic maps aligned to TMDM.

2 Design Objective

Since the release of TMAPI 1.0 several Topic Maps standards have been published, especially the Topic Maps Data Model (TMDM [2]) must be emphasized here. Because the initial version of TMAPI does not support all facets of TMDM well, the main design objective for 2.0 was TMDM compliance and the observance of its constraints to some extend.

Due to reasons explained in the introduction, TMAPI 2.0 is explicitly Java-centric and requires Java 1.5 since it utilizes generics and variable arguments; translations to other programming languages should be handcrafted to account for respective language specifics. The UML class diagrams for the core and the index package provided by the TMAPI project can serve as starting points for translations to other object-oriented programming languages.

While the first version does not offer any filtering methods (i.e. iterating over the occurrences of a topic by the occurrence's type), the second generation provides simple filters to ease the development of applications. A more advanced filter language was rejected for the time being but may find its way into a subsequent release.

3 Status

The project members have published UML class diagrams which describe the current status of the project. In favour of readability the class methods are omitted.

These UML class diagrams were used as boilerplate for the project's interfaces.

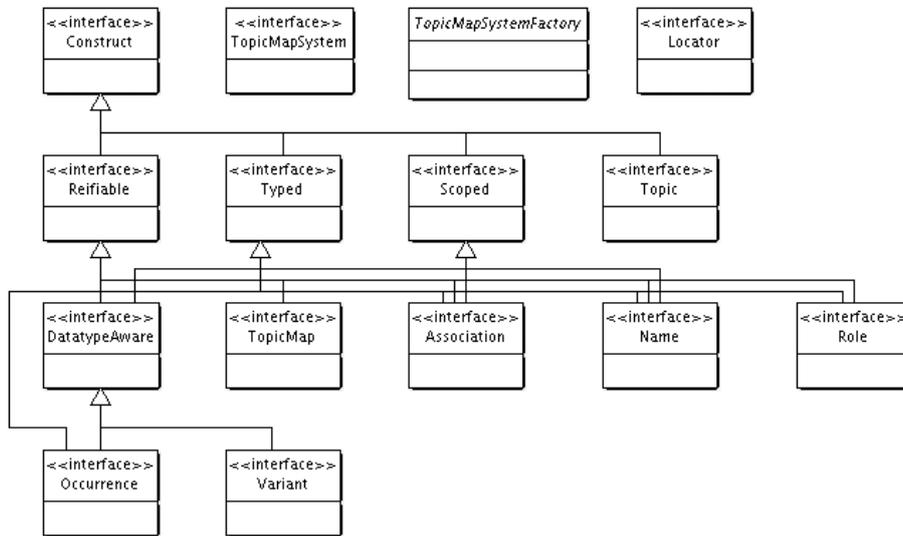


Fig. 1. Abbreviated UML class diagram for the "org.tmap.core" package

While the first TMAPI version offers just 89 tests to ensure compliance, the new release will provide a suite with approximately 250 tests. The enhanced test suite ensures that different implementations are conform to certain requirements and establishes a profound basis for application programmers to test particular Topic Maps processors against. Further, these tests corroborate the claim that applications which use the project’s interfaces are portable over different Topic Maps processors.

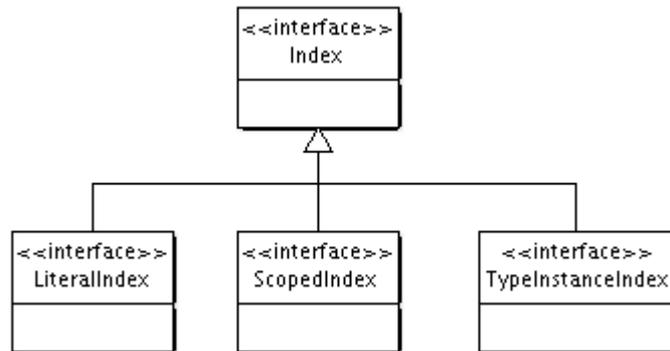


Fig. 2. Abbreviated UML class diagram for the "org.tmapl.index" package

4 Changes

The following sections enumerate important changes between TMAPI 1.0 and 2.0.

4.1 Changes in core

TMAPI 2.0 introduces several generalized interfaces like *Reifiable*, *Typed*, *Scoped*, and *DatatypeAware*. These interfaces avoid redundant method declaration (i.e. *setType()/getType()*, *setValue()/getValue()*, et al.).

Additionally, the *ConfigurableHelperObject* was eliminated since it was only utilized by the *Index* interface. The indices are now available by simply calling *TopicMap.getIndex(Class indexInterface)*.

As mentioned above one objective was to enforce TMDM constraints. Thus TMAPI 2.0 is more restrictive than its predecessor concerning model constraints (i.e. disallows *Role.setPlayer(null)*).

The naming in TMAPI 2.0 is simplified for convenience:

- *TopicName* is called *Name*
- *AssociationRole* is called *Role*
- *Topic Maps construct* is called *Construct* (TMAPI 1.0's equivalent is *TopicMapObject*)

4.2 Changes in index

The main changing covers the reduction to only three indices:

- *TypeInstanceIndex*
- *ScopedIndex*
- *LiteralIndex*

This approach distances from a single construct view to a generalized view on a topic map ("literal view", "typed view", and "scoped view"). From these views specific constructs can be accessed (i.e. return all associations in scope x). Topic Maps constructs are available in multiple indices, i.e. *Occurrence* in *TypeInstanceIndex*, *ScopedIndex*, and *LiteralIndex*. The reduction to three indices makes reindexing and / or synchronization more expensive: I.e. a *TypeInstanceIndex.reindex()* operation has to resynchronize the information about topics, associations, roles, occurrences, and names, while a TMAPI 1.0 *AssociationsIndex.reindex()* would only update the information about associations. However the project members believe that *Index* implementations will rather realize constant synchronization.

The *IndexFlags* interface was abolished. Its only method *isAutoUpdated()* is now available in the *Index* interface.

4.3 Specific changes

DatatypeAware Is the superinterface for *Occurrence* and *Variant*. Therefore it provides several methods for value assignments. It requires the Topic Maps processor to set the datatype implicitly to *xsd:string* in *setValue(String value)* and to *xsd:anyURI* in *setValue(Locator value)*. For convenience, it offers several methods to set and read values where the datatype is implicitly assigned and introduces *setValue(String value, Locator datatype)* in order to be consistent with TMDM's concept of datatypes; *getDatatype()* returns the *Locator* identifying the datatype of the value.

Topic Provides filter methods *getRolesPlayed(Topic type)*, *getNames(Topic type)*, *getOccurrences(Topic type)* which return only those constructs which have the specified type. Further, various factory methods for *Name* and *Occurrence* are provided, inter alia a method for creating names with the default name type.

Association Does not allow `null` for player and type assignments. Further, *getRoleTypes()* and a method to filter the association roles is provided.

Role Does not allow to set the role player and type to `null`.

TopicMap Provides *getTopicBySubjectIdentifier()* and *getTopicBySubjectLocator()* (moved from the index package). Even more importantly, the TopicMap interface does not allow to create topics without any identity, such as an item identifier, a subject identifier, or subject locator.

5 Conclusions and Further Work

The project is currently in alpha status but it should have reached a certain degree of maturity when this paper gets published. TMAPI 2.0 benefits from the meanwhile finalized TMDM. While the previous version supports the XTM 1.0 model [1] and some aspects of TMDM, TMAPI 2.0 has shifted to a TMDM compliant API which also considers programmers' convenience requirements.

Some interesting proposals, like a more advanced filter language or interfaces for TMQL, have been delayed due to lack of human resources and time. Further, TMAPI lacks of a standardized transaction management which seems to be necessary prior TMAPI gets accepted in an enterprise context.

The remaining paper elaborates on the rejected advanced filter mechanism which is meant to bridge a gap between a complete query language and a programming API.

5.1 Filter Language

Even if TMQL is close to be an ISO standard, the success of Microsoft's LINQ [4] and the recent popularity of domain-specific languages [6] has shown that there is desideratum to have specialized languages available which solve particular problems. Ideally, the developer can stay in the familiar programming language.

The new TMAPI version supports some limited filter methods like navigating from a topic to its occurrences which have a particular type, but these filter methods are not satisfactory for more complex tasks like navigating to all occurrences with a particular type and returning the value if the datatype is *xsd:string*. To accomplish such a navigation, the application developer has to write code against TMAPI which might be tedious or she has to switch to another language like TMQL which requires some learning effort.

A simple, domain-specific filter language should be a good, intermediate solution here: The developer stays in her familiar programming language and uses the usual tools and can utilize type checking performed by the compiler.

Due to lack of resources the filter proposal has not been worked out completely, but the general idea is, that the TMAPI project would provide a new, immutable interface *Filter* which can be passed around to all kind of interfaces which represent a particular Topic Maps construct.

One possibility to create such a *Filter* would be the mentioned domain-specific language:

```
// Return those role players which play the role "group" in a
// "member-of" association where the current topic plays the
// role "member":

Filter<Topic> filter =
    roles(member).parent(memberof).roles(group).select(player);

for (Topic player: topic.match(filter)) {
    doSomethingWith(player);
}
```

The language used to create the filter should be obvious: The filter takes the current topic as context to navigate to the played roles and compares the role type with the topic "member". For each role the parent association is visited and its type is compared to the topic "member-of". From the association, the filter navigates down to each role of type "group" and selects the player from it.

Even though the domain-specific language leaves room for improvement, the equivalent TMAPI code is certainly longer:

```
// Visit all role the topic plays
for (Role r: topic.getRolesPlayed()) {
    if (!r.getType().equals(member)) {
        continue;
    }
    Association assoc = r.getParent();
    // Compare the association's type
    if (!assoc.getType().equals(memberof)) {
        continue;
    }
    for (Role role: assoc.getRoles()) {
        if (role.getType().equals(group)) {
            doSomethingWith(role.getPlayer());
        }
    }
}
}
```

Due to the immutability of *Filter* it can be reused in several contexts, while the code on top of TMAPI is not easily reusable unless the developer creates a library for common tasks.

Since not every TMAPI implementation has the necessary resources, the project itself should provide a generic implementation. This default implementation would therefore work with every TMAPI compatible implementation, even if it might not be optimized for the specific Topic Maps processor.

A "service provider interface" would enable TMAPI implementations to provide Topic Maps processor-specific, optimized implementations of the *Filter*.

The authors of this paper regard the filter language with a default implementation as reasonable extension to the current interfaces since it provides rich navigation facilities and reduces development time considerably.

References

1. ISO/IEC. 13250:2003: Information Technology — Document Description and Processing Languages — Topic Maps. Technical report, International Organization for Standardization, Geneva, Switzerland., 2003.
<http://www.y12.doe.gov/sgml/sc34/document/0322files/iso13250-2nd-ed-v2.pdf>.
2. ISO/IEC. IS 13250-2:2006: Information Technology — Document Description and Processing Languages — Topic Maps — Data Model. Technical report, International Organization for Standardization, Geneva, Switzerland., 2006.
<http://www.isotopicmaps.org/sam/sam-model/2006-06-18/>.
3. ISO/IEC. FCD 18048: Information Technology — Document Description and Processing Languages — Topic Maps — Query Language (TMQL) 2008-05-15. Technical report, International Organization for Standardization, Geneva, Switzerland., 2008.
<http://www.isotopicmaps.org/tmql/tmql.html>.
4. Microsoft Corporation. The LINQ Project, 2005.
<http://msdn.microsoft.com/enus/library/aa479865.aspx>.
5. J. Schmidt. PHPTMAPI.
<http://phptmapi.sourceforge.net/>.
6. D. Spinellis. Notable design patterns for domain specific languages. *Journal of System and Software*, 56(1):91–99, Feb. 2001.
7. TMAPI project. Topic Maps API.
<http://www.tmap.org/>.
8. TMAPI4NET project. TMAPI for .NET.
<http://code.google.com/p/tmap4net/>.

TMCL and OWL

Lars Marius Garshol

Bouvet, Oslo, Norway

larsga@bouvet.no

Abstract. This paper compares the Topic Maps schema language TMCL with the corresponding RDF technologies RDFS/OWL, and describes the first method for bidirectional conversion between TMCL and RDFS/OWL, based on an existing RDF-to-TM mapping for instance data. The conversion from TMCL creates OWL Lite ontologies where possible, and OWL DL ontologies where not.

1 Introduction

Today, it is possible to convert instance data from RDF to Topic Maps, and vice versa [Garshol03a], and it is even possible to use the same vocabulary in both technologies. However, it has not been possible to take a vocabulary description in RDFS or OWL and convert this into a TMCL schema [TMCL]. Nor has conversion in the opposite direction been possible.

The result is that anyone wishing to use a vocabulary defined in one technology with the other is forced to translate the schema (or ontology description) manually, in order to be able to use tools such as schema-driven editors, validators, reasoners, and so on. Such work is tedious, error-prone, and also requires users to know both technologies quite intimately.

With a reliable conversion method implemented in tools, migration is dramatically simplified for users, who no longer need to learn three schema languages (RDFS, OWL, and TMCL), being able instead to simply use the editor of their choice. The schema conversion problem has so far been unsolved, despite some early work on OWL and Topic Maps interoperability, described in section 6.

This paper presents a bidirectional conversion method, which it claims effectively solves the schema conversion problem. The conversion method is

based on existing RDF-to-TM and TM-to-RDF mappings, in such a way that valid instance data, once converted, will also validate according to the converted schema. This ensures that the instance and schema conversion methods work well together.

1.1 The RTM mapping

RTM is a conversion method from RDF to Topic Maps [Garshol03b]. It is based on the observation that resources¹ in RDF correspond to topics in Topic Maps, while statements correspond to names, occurrences, or associations. The RTM mapping vocabulary, which is an RDF vocabulary for describing the mapping from RDF to Topic Maps of a particular RDF vocabulary, is needed because RDF statements do not contain sufficient information to determine which of the three Topic Maps constructs they should be mapped to.

The basic workings of the mapping can be summarized as follows:

- Resources become topics, and their URIs become subject identifiers.
- Statements become names, occurrences, or associations, and which is determined by a mapping attached to the RDF property.
- Association roles are fixed for properties mapped to associations, by specifying one role type for the subject of the statement, and one for the object.

Below is a simple example mapping for three properties from Dublin Core, expressed in n3[Berners-Lee05], just to show the basic approach used:

```
dc:title rtm:maps-to rtm:basename .
dc:date rtm:maps-to rtm:occurrence .
dc:creator rtm:maps-to rtm:association .
dc:creator rtm:subject-role resource .
dc:creator rtm:object-role value .
```

Given this, the following RDF graph (in n3):

```
<#tmcl-owl> dc:title "TMCL and OWL";
             dc:date "2008-09-15";
             dc:creator <#lmg> .
```

would be converted to the following topic map (in CTM):

¹ The RDF terminology does not match that of Topic Maps entirely here, as resources and nodes are conflated in RDF, and so an exact match with the subject/topic distinction in Topic Maps is not possible.

```
tmcl-owl - dc:title "TMCL and OWL";  
          dc:date: "2008-09-15" .  
dc:creator(resource : tmcl-owl; value : lmg)
```

2 Comparing TMCL with RDFS/OWL

TMCL is the standard used in the Topic Maps family of technologies to describe the proper use of a vocabulary. On the RDF side, there are two corresponding standards: RDFS and OWL. The relationship between these is that OWL is an extension of RDFS, providing additional facilities.

The structure of TMCL, RDFS, and OWL is mostly similar, in that all are vocabularies allowing the host technology to describe itself. That is, TMCL is a Topic Maps vocabulary, just as RDFS and OWL are RDF vocabularies. TMCL goes a bit further, however, and also allows constraints to be defined in a Schematron-like way, using TMQL expressions.

The vocabulary elements of these three languages can be roughly divided into three groups:

- Constraints. These are rules describing the structure of instance data.
- Documentation. This is information meant for human readers, such as names, descriptions, version information, and so on.
- Reasoning rules, which is information whose only use is to allow further information to be inferred from existing instance data.

RDFS and OWL both have documentation vocabulary elements, while TMCL does not, which means that such elements must be converted by the RTM mapping as is, in the hope that Topic Maps software will display it to humans. Similarly, pure reasoning statements can be converted as is, in the hope that some software will make use of it, although this can by no means be guaranteed.

The conversion rules generally attempt to express all RDFS and OWL constraints by means of the TMCL core constraints. This is not possible in all cases, but the remaining elements can be expressed using TMQL constraints. TMQL constraints allow the validation behaviour to be captured, but the ability of software to introspect the constraints and use the information for other purposes (such as schema-driven editing) is lost.

2.1 Validation versus reasoning

The general purpose served by these languages is the same: to allow users to describe the proper use of their vocabularies. However, the approach taken by TMCL is very different from that taken in RDFS and OWL. Generally, one could say that while TMCL has validation semantics, RDFS and OWL have inference semantics.

To give an example: it is possible to say, in both TMCL and RDFS, that `dc:creators` must be persons. However, the effect of encountering something which is a `dc:creator`, but not a person is different. In TMCL this is a validation error; it is assumed that the data is wrong (either the topic is lacking a type, or the topic should not appear in this association). In RDFS this is not an error per se; instead, it is treated as a case of missing data, and the reasoner assumes that the resource is a person after all.

OWL heavily emphasizes the use of logical reasoning, to the extent of having a direct mapping into Description Logic, a class of formal (that is, mathematical) logical languages. Reasoning on OWL datasets is thus underpinned by description logic, which guarantees that reasoning will be efficient. Strictly speaking, OWL is divided into three subsets: Lite, DL, and Full, where only the former two can be mapped to description logic. (OWL Full is known to be undecidable.)

It should be emphasized that this is a difference of purpose more than an essential difference. It is possible to use TMCL for reasoning, even if no reasoning semantics are given in the standard. Similarly, it is possible to do some validation with RDFS and OWL through negative statements like making classes disjoint. Alternatively, one could devise one's own validation semantics for RDFS/OWL schemas.

2.2 Consequences for the conversion

That TMCL has validation semantics, while RDFS/OWL have inference semantics complicates the creation of conversions between them, since in general creating schemas that are treated the same way in the two technologies is impossible, given that in one they are used for validation and in another for inference. The conversion method presented in this paper ignores this issue, and instead aims to convert structural information to the nearest possible equivalent in the target technology.

To stay with the example above: if an RDFS schema says that `dc:creators` must be `persons` that particular statement is easily reproduced in TMCL. That TMCL

will treat this information differently from how RDFS will treat it is left as an issue for the user to handle.

An alternative approach might have been to define a reasoning semantics for a TMCL extended with the necessary OWL elements, which would have allowed RDFS/OWL ontologies to be converted into structures with the exact same meaning on the Topic Maps side. However, the general requirement of Topic Maps users is for validation much more than for reasoning, and so this approach was not taken.

3 RDFS to TMCL conversion

Any attempt at conversion has to begin with RDFS, which describes the basic structure of ontologies. The core of RDFS is essentially three vocabulary elements: `rdfs:Class`, `rdfs:range`, and `rdfs:domain`. `rdfs:Class` is the class of all classes, and thus equivalent to `tmcl:topicitype`. That is, instances of `rdfs:Class` are classes, like `person`, `organization`, `country` etc.

Which properties an instance of a class can have are defined by `rdfs:domain`, which relates an RDF property to the classes which can be the subjects of statements with this property. Similarly, for these properties `rdfs:range` gives the classes which can be the values (or objects) of statements with these properties. If the property values are literals, the class `rdfs:Literal` (or a subclass of it) is used.

Conversion of `rdfs:Class` is straightforward: instances of this class become instances of `tmcl:topicitype`. This leaves the constraints on properties, which must be converted using a special algorithm, as the mappings here are more complex. The algorithm mapping rules are given below, using a simple RDF triple filter syntax in the GIVEN section and CTM notation in the OUTPUT section.

```
GIVEN x rdfs:domain y &
      x rtm:maps-to rtm:basename
OUTPUT
x isa tmcl:nametype .
?c isa tmcl:topicname-constraint .
tmcl:applies-to(tmcl:constraint-role : ?c,
               tmcl:topicitype-role : y)
tmcl:applies-to(tmcl:constraint-role : ?c,
               tmcl:nametype-role : x)
```

This mapping converts properties which map to topic names into a name type in TMCL, and for each `rdfs:domain` statement creates a corresponding constraint

attaching the property to the given class. It is an error for the range to be something other than `rdfs:Literal` or `xsd:string`.

```
GIVEN x rdfs:domain y &
      x rtm:maps-to rtm:occurrence
OUTPUT
x isa tmcl:occurrencetype .
?c isa tmcl:topicoccurrence-constraint .
tmcl:applies-to(tmcl:constraint-role : ?c, tmcl:topicity-
role : y)
tmcl:applies-to(tmcl:constraint-role : ?c,
                tmcl:occurrencetype-role : x)
```

This mapping is the same as the previous one, except that it deals with occurrence types.

```
GIVEN x rdfs:range z &
      x rtm:maps-to rtm:occurrence
OUTPUT
?c isa tmcl:occurrencedatatype-constraint
tmcl:datatype: z .
tmcl:applies-to(?c : tmcl:constraint-role,
                x : tmcl:occurrencetype-role)
```

This mapping turns range statements for occurrence types into datatype constraints. (This assumes that the range really *is* a datatype. If it is not, conversion software should treat this as an error.)

```
GIVEN x rdfs:domain y &
      x rtm:maps-to rtm:association &
      x rtm:subject-role s
OUTPUT
x isa tmcl:associationtype .
s isa tmcl:roletype .

?c isa tmcl:associationrole-constraint
tmcl:card-min: 1
tmcl:card-max: 1 .
tmcl:applies-to(tmcl:constraint-role : ?c,
                tmcl:assoctype-role : x)
tmcl:applies-to(tmcl:constraint-role : ?c,
                tmcl:roletype-role : s)

?c2 isa tmcl:roleplayer-constraint
tmcl:applies-to(tmcl:constraint-role : ?c2,
                tmcl:assoctype-role : x)
tmcl:applies-to(tmcl:constraint-role : ?c2,
                tmcl:assoctype-role : s)
tmcl:applies-to(tmcl:constraint-role : ?c2,
                tmcl:assoctype-role : y)
```

This mapping handles domain statements for association types, marking the association type and subject role type as the correct kind of type, then creating the constraints to connect the role type to the association type and to the topic type that the domain is converted to.

```
GIVEN x rdfs:range z &
      x rtm:maps-to rtm:association &
      x rtm:object-role o
OUTPUT
x isa tmcl:associationtype .
o isa tmcl:roletype .

?c isa tmcl:associationrole-constraint
  tmcl:card-min: 1
  tmcl:card-max: 1 .
tmcl:applies-to(tmcl:constraint-role : ?c,
                tmcl:asstype-role : x)
tmcl:applies-to(tmcl:constraint-role : ?c,
                tmcl:roletype-role : o)

?c2 isa tmcl:roleplayer-constraint
tmcl:applies-to(tmcl:constraint-role : ?c2,
                tmcl:asstype-role : x)
tmcl:applies-to(tmcl:constraint-role : ?c2,
                tmcl:asstype-role : o)
tmcl:applies-to(tmcl:constraint-role : ?c2,
                tmcl:asstype-role : z)
```

This mapping is the same as the previous one, except that it handles the range. Cardinalities are not specified with RDFS, but can be inferred from OWL, as in section 4.2. Symmetric association types will also not be converted correctly here, as this conversion assumes that the subject and object roles are different. Section 4.4 shows how to convert such properties correctly.

The rest of the RDFS vocabulary can be handled by the RTM mapping, as follows:

- `rdfs:subClassOf` is mapped to TMDM superclass-subclass associations.
- `rdfs:subPropertyOf` is mapped to TMDM superclass-subclass associations.
- `rdfs:label` is mapped to a base name with the default type.
- `rdfs:comment` is mapped to an occurrence of type `rdfs:comment`.
- `rdfs:seeAlso` is mapped to an occurrence of type `rdfs:seeAlso`.
- `rdfs:isDefinedBy` is mapped to an association of type `tmcl:definedBySchema`².

² This association type has disappeared in the latest TMCL draft, but may return.

4 OWL to TMCL conversion

While RDF Schema is a relatively straightforward schema language describing constraints on vocabularies, OWL is a rather more complicated language which does not translate so easily into vocabulary constraints, since the emphasis is much more on supporting inferencing. The abstract syntax is also much more involved, which greatly complicates conversion.

4.1 Class and property relationships

A central part of OWL is the ability to define classes in terms of their relationships with other classes. For example, one can state that one class has no instances in common with another, or that a class is the intersection of two other classes, and so on.

`owl:disjointWith` is easily convertible, since TMCL has exactly the same capability:

```
GIVEN x owl:disjointWith y
OUTPUT
  ?c isa tmcl:exclusive-instance .
  tmcl:applies-to(tmcl:constraint-role : ?c,
                  tmcl:topic-type-role : x)
  tmcl:applies-to(tmcl:constraint-role : ?c,
                  tmcl:topic-type-role : y)
```

`owl:complementOf` states that the instances of a class is exactly those topics which are not instances of the other class. This cannot be expressed in TMCL. However, it is merely a stronger form of `owl:disjointWith`, in the sense that the statement `A owl:disjointWith B` divides the universe of instances into three disjoint sets: instances of A, instances of B, and those which are instances of neither.

The statement, `A owl:complementOf B`, however, says that all instances which are not instances of B are instances of A, and vice versa. This strengthens the previous statement by saying that the last set (those which are instances of neither) is empty.

In TMCL, this can be expressed with the same `tmcl:exclusive-instance` constraint, and an additional TMQL constraint which makes it an error for instances not to be an instance of either of the two classes.

`owl:intersectionOf` states that the instances of a class is exactly the intersection of the extensions of a set of other classes. That is, every instance of

the class is an instance of all the other classes. This can be expressed with a TMQL constraint.

`owl:unionOf` states that the instances of a class is exactly the union of the extensions of a set of other classes. That is, every instance of the class is an instance of at least one of the other classes. This can be expressed with a TMQL constraint.

OWL has two properties for stating that two classes or properties have the same set of instances (extension), but that their meanings (intentions) are different. This can be expressed indirectly in Topic Maps by creating a subtyping loop:

```
GIVEN c1 owl:equivalentClass c2
OUTPUT
  tmdm:superclass-subclass(tmdm:superclass : c1,
                           tmdm:subclass : c2)
  tmdm:superclass-subclass(tmdm:superclass : c2,
                           tmdm:subclass : c1)

GIVEN p1 owl:equivalentProperty p2
OUTPUT
  tmdm:superclass-subclass(tmdm:superclass : p1,
                           tmdm:subclass : p2)
  tmdm:superclass-subclass(tmdm:superclass : p2,
                           tmdm:subclass : p1)
```

4.2 Cardinalities

Some constraints in OWL ontologies must be specified in a rather unusual way. First, `owl:Restriction` is used to define a nameless class which specifies the restriction on a particular property (indicated with `owl:onProperty`). Second, the class you wish to constrain is related to the nameless class, usually by making the constrained class a subclass of the restriction. This makes the constraint also apply to the desired class. In OWL, (some kinds of) cardinalities and the tightened restrictions on allowed property values must be specified in this way.

Cardinalities can be expressed with `owl:minCardinality` and `owl:maxCardinality`, or with `owl:cardinality`, which is a shorthand for setting both to the same value. However, it's also possible to set cardinality on the property itself without using a restriction, simply by making the property an instance of `owl:FunctionalProperty`, which states that it is inherent to the property itself that it can have no more than one value per resource.

So, finding the minimum cardinality of property `p` on class `c` becomes rather involved, but with

```
GIVEN c rdfs:subClassOf r,
      r rdf:type owl:Restriction,
      r owl:onProperty p,
      r owl:minCardinality m
```

we can assume that the minimum cardinality is m . If this yields nothing, replace `owl:minCardinality` with `owl:cardinality` and try again. If the result is nothing, repeat for all superclasses and choose the largest value. If nothing is found, assume 0.

For maximum cardinality the procedure is the same, except that if the property is an instance of `owl:FunctionalProperty` the maximum is 1. If nothing is found, assume infinity.

The cardinalities produced by this process can be inserted in the constraints produced by the RDFS-to-TMCL conversion algorithm defined earlier to make it more accurate.

4.3 Value restrictions

Another use of restrictions in OWL is to specify constraints on the possible values of a property. The three properties which can be used for this purpose are described below.

`owl:allValuesFrom` states that all values of a given property in a certain class must be of a certain class. This is effectively what `rdfs:range` does, but `owl:allValuesFrom` is used to narrow the range of a property on a specific class.

`owl:someValuesFrom` states that some values of a given property on a certain class must be of a certain class. That is, at least one value must be of that class. This is like the previous property, except that values of other classes are also allowed.

`owl:hasValue` states that instances of a class must have a specific value for a specific property. Other values are allowed, but this one specific instance must be among the values.

None of these three properties can be expressed directly in TMCL; however, they can be expressed with TMQL constraints.

4.4 Input to mapping

Some OWL vocabulary elements are, strangely, not applicable in Topic Maps, but can be used to infer the possible settings of the RTM mapping. This is due to

the structural differences between RDF and Topic Maps, which means that some schema information that is relevant in RDF quite simply does not apply in Topic Maps.

The `owl:inverseOf` property says that one property `p2` is the inverse of another property `p1`. From a Topic Maps point of view this means that both `p1` and `p2` must be association types, and further that they must be the *same* association type, but with the role types reversed. (That is, the subject role of `p1` is the object role of `p2`, and vice versa.) This is easily expressed using the RTM mapping.

The `owl:DatatypeProperty` is a class of properties whose values are literals, and implies that the property must map to a name or an occurrence in Topic Maps. Similarly, `owl:ObjectProperty` instances must have resource values, and must map to an occurrence or an association in Topic Maps.

`owl:SymmetricProperty` is a property class which implies that the property is symmetric, meaning that `x p y` implies `y p x`. This means that the property must be an association type in Topic Maps, and that the subject and object roles must be the same. It also means that the conversion of the association type must be a little different from that given for RDFS:

```
GIVEN: x rdfs:domain y &
       x rdfs:range z &
       x rtm:maps-to rtm:association &
       x rtm:subject-role r &
       x rdf:type owl:SymmetricProperty

OUTPUT:
x isa associationtype .
r isa roletype .

?c isa associationrole-constraint
  card-min: 2
  card-max: 2 .
applies-to(constraint-role : ?c, associationtype-role : x)
applies-to(constraint-role : ?c, roletype-role : r)
```

Associations expressed in this way will automatically be treated as symmetric by Topic Maps software, while in RDF this requires reasoning. However, the OWL ontology expresses the symmetric nature of the relationship much more directly than the TMCL schema does. It might be, therefore, that the `owl:SymmetricProperty` class should be converted to Topic Maps, to preserve this information. Or, perhaps, that TMCL should be extended to allow this to be expressed in TMCL.

4.5 Documentary information

The documentary parts of OWL can be handled straightforwardly by the RTM mapping, as these are structurally simple and are not required to match any particular structure on the Topic Maps side.

The vocabulary elements in question are:

- `owl:DeprecatedClass` is the class of deprecated classes and can be converted as is.
- `owl:DeprecatedProperty` is the class of deprecated properties and can be converted as is.
- `owl:versionInfo` becomes an occurrence of the same type, containing the same version information.
- `owl:priorVersion` becomes an association of the same type in Topic Maps, relating the schema to another schema.
- `owl:backwardCompatibleWith` becomes an association of the same type in Topic Maps, relating the schema to another schema.
- `owl:incompatibleWith` becomes an association of the same type in Topic Maps, relating the schema to another schema.
- `owl:Schema` corresponded to `tmcl:Schema` in earlier TMCL drafts, but this has disappeared in the latest draft. It's not yet clear whether it might return.

4.6 Various elements

OWL contains three classes which there is no particular need to convert:

- `owl:Class` is the same as `rdfs:Class`.
- `owl:Thing` is the same as `rdfs:Resource`.
- `owl:Nothing` is the empty class. As such it can be converted directly into Topic Maps. The semantics can be reproduced with a TMQL constraint, though it is difficult to see what purpose this could possibly serve.

`owl:sameAs` is used to say that two instances, properties, or classes are the same, but without merging them. There is no way to do this in Topic Maps, but in Topic Maps it is possible to merge topics without losing identifiers, and so the two topics can simply be merged. This does lose the distinction between the two topics, thus losing information, and so an alternative would be to convert the statement to an association of type `owl:sameAs`. However, as this association type is not supported by, for example, TMQL engines, the former approach is preferable.

`owl:InverseFunctionalProperty` is the class of all properties for which every value must be unique. That is, two different resources cannot have the same value for an inverse functional property. In the current TMCL draft this is only expressible for occurrence types, as follows:

```
GIVEN p rdf:type owl:InverseFunctionalProperty &
      p rtm:maps-to rtm:occurrence &
      p rdfs:domain c
OUTPUT
?c isa tmcl:uniqueoccurrence-constraint .
tmcl:applies-to(tmcl:constraint-role : ?c,
               tmcl:topic-type-role : c)
tmcl:applies-to(tmcl:constraint-role : ?c,
               tmcl:occurrence-type-role : p)
```

For names and associations the constraint must be expressed with TMQL constraints.

`owl:differentFrom` and `owl:AllDifferent` is like `owl:disjointWith`, except it states that individuals are different. This is expressible with TMQL constraints.

`owl:oneOf` is used to state that the instance set of a class is closed. It can also be combined with `owl:DataRange` to define a datatype as a set of values. This is not possible in TMCL, but is easily replicated with a TMQL constraint.

4.7 Non-convertible elements

The following OWL vocabulary elements cannot be converted:

- `owl:TransitiveProperty` is only used for reasoning and so has no equivalent in TMCL. However, it is a simple topic type, and so can be converted as is, to be used by systems which understand it.
- `owl:imports` is an import mechanism, and as such best handled by doing the import before conversion to TMCL.
- `owl:OntologyProperty` is a class of properties which simply must relate an `owl:Ontology` to another. There is no real need to convert this, as the same information will be expressed in RDFS.
- `owl:AnnotationProperty` is a class of properties for which no constraints (“property axioms”) are allowed in OWL DL. It is up to the user to decide which properties are annotation properties, and which are not. This ensures that ontologies remain within what can be expressed in description logic. It carries no special semantics, and so there is no need to convert it to Topic Maps.

5 TMCL to RDFS/OWL conversion

The TMCL to RDFS/OWL conversion relies on the TMR vocabulary [Garshol03c] for describing mappings from Topic Maps to RDF, and produces an RDFS/OWL schema for the RDF models that result from the conversion. The conversion is designed to produce ontologies that conform to OWL Lite where possible, and OWL DL where it is not. The conversion will produce OWL Lite ontologies as long as the TMCL schema has no abstract classes and no exclusive instance constraints. It will always produce OWL DL ontologies, as long as all the `tmcl:*type` topic types are disjoint, and as long as topics do not appear both as types and as instances.

No attempt is made to convert TMQL constraints, even though in many cases it may be that the TMQL constraints might have been expressible with OWL, especially if they have been converted from RDFS/OWL in the first place. TMQL constraints are very difficult to introspect, and so this was considered out of scope.

5.1 The TMR mapping

The TMR mapping vocabulary essentially provides two pieces of information for the conversion. One is what RDF property type to convert names of the default type to for a given topic type. The other is which of the roles in a binary association to turn into the subject of the resulting RDF statement, and which to turn into the object.

A TMR mapping of the Dublin Core example in 1.1 is shown below in CTM:

```
tmr:name-property(tmr:type : dcc:resource,
                 tmr:property : dc:title)
tmr:preferred-role(tmr:association-type : dc:creator,
                  tmr:role-type : dcc:resource)
```

5.2 The conversion

The actual conversion is rather complicated, and so no attempt is made to define it formally in this paper. However, the general gist of it can be given rather quickly.

The TMR conversion is used for elements not explicitly mentioned her, and so the names of all topics which become properties and classes turn into

`rdfs:labels`. Similarly, documentary information, such as comments etc are converted using the usual TMR conversion.

Every `tmcl:topicType` becomes an `owl:Class`. Subclassing associations are turned into `rdfs:subClassOf` statements.

Every name type becomes an `owl:DatatypeProperty`. This includes both those defined as name types, and those which replace the default name type in the TMR mapping. If the name type applies to exactly one topic type, that topic type is set as the `rdfs:domain` of the property. Otherwise, no domain is specified. The `rdfs:range` is always set to `rdfs:Literal`.

Occurrence types are treated the same way, except that there is no mapping of types with TMR. `rdfs:range` is set to the specified data type, if any, and otherwise defaults to literal.

Association types are rather more difficult. If they are n-ary (for n larger than 2) associations become resources, and the schema must be converted accordingly. If they have only a single role type, with an association role constraint stating that cardinality of the role type is exactly 2, then the association type is symmetric. It then becomes an instance of `owl:SymmetricProperty` and domain and range assume the same value.

For binary associations, the conversion is easier. They are specified as `owl:ObjectProperty`. One role type is either specified as the preferred role type, or one is chosen at random. If only one topic type can play this role type, that becomes the domain. Handling of the other role type is the same, except that the topic type becomes the range.

The cardinalities of names, occurrences, and associations on various topic types can be expressed easily with `owl:Restrictions`.

`tmcl:exclusive-instance` constraints can be mapped directly into `owl:disjointWith` statements.

5.3 Abstract classes

The abstract topic type constraint states that a particular topic type cannot have instances which are direct instances of that type, but that they must instead be instances of a subclass of it. This is difficult to express in OWL, since OWL does not distinguish between statements which are made directly and those which are inferred.

However, a kind of solution is possible³, by stating that the abstract class is `owl:equivalentClass` to the `owl:unionOf` its subclasses. This would make it impossible for a resource to be an instance of a the class without being an instance of its subclasses.

5.4 Non-convertible elements

Some parts of TMCL cannot be converted because no corresponding constraints exist in RDFS and OWL. These are:

- Constraints on subject identifiers and subject locators.
- Regular expression constraints on string values.
- Scope constraints.
- TMQL constraints (as mentioned above).

Unique occurrence constraints are not convertible, because in TMCL these state that occurrence values are unique within a particular topic type. Since this allows the same value to occur with a different topic type, it is not possible to state that the occurrence type is an `owl:InverseFunctionalProperty`. One could make this assertion if the occurrence type is legal for only one topic type, but this is clearly not safe. It may be that TMCL will change on this point.

6 Related work

Some work has already been published on Topic Maps/OWL interoperability, but unfortunately these efforts have had a different focus.

[Garshol03a] focuses on interchange of instance data, but had some early notes on how parts of OWL could be translated into Topic Maps. These were just notes, however, and as RDFS was not covered, the work was not sufficient to allow schema conversion in any real sense.

[Cregan05] mapped the TMDM to an RDF vocabulary, representing every item type as an RDF class, and every property as an RDF property. The resulting vocabulary was then described with OWL. This is what [pepper06] describes as an “object mapping” and therefore deficient as a way of mapping instance data. OWL was only used to describe the RDF vocabulary, and so there was really no mapping from Topic Maps to OWL or vice versa.

³ I am indebted to David Norheim for this solution.

[Vatant04] proposed describing Topic Maps ontologies in OWL, using a small extra Topic Maps vocabulary. Strictly speaking, this was not work on Topic Maps/OWL interoperability, but more a suggestion for how to create Topic Maps schemas at a time when TMCL was not available. [Vatant03] is a precursor to this work, and provides an earlier view of the same ideas.

The OMG's Ontology Definition Metamodel specification [ODM] contains a mapping from Topic Maps to OWL, which really is a mapping to RDF and OWL. The mapping essentially infers an ontology from the structure of the topic map. So every topic which has at least one instance, or participates in a superclass-subclass association, becomes an `owl:Class`, for example. As such, it does provide as much interoperability on the schema level as was possible before the introduction of TMCL. However, since it was published before TMCL, TMCL schemas are not covered at all.

7 Conclusion and further work

The conversion method presented here has been prototypically implemented in Jython on top of the Ontopia Knowledge Suite and Jena. Several existing ontologies have been converted in both directions, allowing the conversion method to be verified, and results have been satisfactory on RDFS/OWL to TMCL conversions.

In the opposite direction it has been found in many cases that name types, occurrence types, and association types can often apply to more than one topic type. In these cases the domain and range cannot be expressed with RDFS, and so part of the schema's structure is lost. This can be corrected in RDFS tools by manually introducing new common superclasses, but giving these correct URIs and names is not possible automatically.

Generally, what remains is to implement the conversion methods in the OKS and to gain more experience with them to see how well they work in practice. Ideally, the Ontopoly Topic Maps editor should start using TMCL instead of the current Ontopoly-specific schema language, and support for these conversion methods should be added to Ontopoly.

TMCL is also not yet finished, and so the conversion method presented here must be updated as future changes occur. It may also be that some of the concerns expressed in this paper will motivate changes to TMCL.

References

- [Cregan05] Cregan, A.: *Building Topic Maps in OWL-DL*; Proceedings of Extreme Markup Languages 2005;
<http://www.idealliance.org/papers/extreme/proceedings/html/2005/Cregan01/EML2005Cregan01.html>
- [Garshol03a] Garshol, L. M.: *Living with Topic Maps and RDF*. Proceedings of XML Europe 2003, 5-8 May 2003, organized by IDEAlliance, London, UK.
<http://www.ontopia.net/topicmaps/materials/tmrdf.html>
- [Garshol03b] Garshol, L. M.: *The RTM RDF to topic maps mapping – Definition and introduction*. Ontopia technical report, 2003-12-28.
<http://www.ontopia.net/topicmaps/materials/rdf2tm.html>
- [Garshol03c] Garshol, L. M.: *TMR: A TM-to-RDF mapping*. Ontopia published subject documentation, 2003-12-17. <http://psi.ontopia.net/tm2rdf/>
- [Berners-Lee05] *Primer: Getting into RDF & Semantic Web using N3*; T. Berners-Lee; World Wide Web Consortium; 2005-08-16; <http://www.w3.org/2000/10/swap/Primer>
- [ODM] *Ontology Definition Metamodel*; Object Management Group; OMG Adopted Specification; November 2007; OMG Document Number: ptc/2007-09-09;
<http://www.omg.org/docs/ptc/07-09-09.pdf>
- [Pepper06] Pepper, S.; *A Survey of RDF/Topic Maps Interoperability Proposals*; W3C Working Group Note 10 February 2006;
<http://www.w3.org/TR/rdfm-survey/>
- [TMCL] *ISO 19756 – Information technology – Topic Maps – Constraint Language*; ISO Committee Draft; 2008-08-07;
<http://www.isotopicmaps.org/tmcl/tmcl.html>
- [Vatant04] Vatant, B.; *Ontology-driven topic maps*; Proceedings of XML Europe 2004;
<http://www.idealliance.org/europe/04/call/xmlpapers/03-03-03.91/.03-03-03.html>
- [Vatant03] Vatant, B.; *OWL and Topic Map Pudding*; Mondeca white paper; 2003-08-01;
<http://web.archive.org/web/20061113154655/http://www.mondeca.com/owl/owltm.htm>

Connecting Information

Building Context Aware P2P Systems with the Shark framework

Thomas Schwotzer

FHTW Berlin (University of Applied Sciences)

thomas.schwotzer@fhtw-berlin.de

Abstract. Abstract: Shark Framework is framework supporting implementation of context aware P2P systems. Shark is an acronym and stand for Shared Knowledge. There is already a theory on context aware P2P systems which is implemented by the Shark framework. Target platforms are in the first step J2SE, J2ME and Android. In next steps iPhone and Microsoft based mobile devices will be supported. Shark FW supports the Knowledge Exchange Protocol (KEP) which is a stateless P2P protocol. Currently KEP has been ported to UDP and TCP. A Bluetooth L2CAP implementation will be available in October. This paper (briefly) explains core concepts of the framework. Sample code illustrates usage of Shark. It is illustrated that just two lines of code are sufficient to set up a peer that exchanges knowledge. This paper is also a call for participation. Shark is an open source project and is open to developers and users.

1 Introduction

Knowledge management is an inherently distributed process. Knowledge is not created in a company, a think tank or whatever. Knowledge is initially created in an individual's mind. Usually people decide to share knowledge. Such newly created knowledge will usually be discussed in groups. It will be exchanged, combined, modified and maybe forgotten. Knowledge always flows within and between groups and between individuals. Groups can be official organizations, e.g. companies but also ad hoc groups or other not official networks of people.

Knowledge management systems (KMS) must support knowledge sharing. Topic maps is a knowledge representation standard. It supports knowledge sharing. The standardized *merge* operation defines how knowledge from different sources can

be integrated. The topic map query language but also the topic maps API can be used to take parts (fragments) from topic maps. Such fragments can be exchanged between topic maps. There are some examples of distributed topic maps applications.

Thus, there are means to combine knowledge but also to take parts of knowledge out of an existing knowledge base that is based on topic maps. Developers of distributed knowledge management systems (more specific: distributed topic maps systems) need additional functionality. A protocol for knowledge exchange is required and applications must decide if and what parts of knowledge are allowed to be exchanged. Currently, this functionality must be completely implemented within the application. Of course, there is no need to implement any marshalling / serialization code line by line. Middleware systems, Web services etc. pp. can be used. Nevertheless, the the whole exchange logic is application specific.

The Shark framework is an open source project. It provides a stateless P2P protocol called KEP (Knowledge Exchange Protocol) and an API for building distributed P2P applications. Shark is independent from a specific knowledge representation format. The Shark concept has its roots in topic maps and therefore most core concepts and ideas are inspired by and compliant to topic maps. A core concept of Shark is the Knowledge Port (KP). A KP can be compared to a TCP or UDP port. It is – hopefully – as easy to define and to open as a socket in e.g. Java. Knowledge ports exchange knowledge particles (more specific: topic map fragments if topic maps engines are used). KPs contain the KEP protocol engine and form the interface between the P2P protocol and the used knowledge base.

The aim of this paper is twofold. First, the Shark framework shall be introduced. It will be shown that an useful P2P communication can already be defined by just a few lines of code.

Secondly, it is a call for participation. There is already a theory on Shark, see e.g. [SG02][MS05] and referenced papers. The Shark framework is new though. Currently it is written in Java and runs with J2SE and J2ME and supports UDP and Bluetooth L2CAP. There are only a few applications yet. Shark will be ported to compact .NET, Android and maybe to iPhone. The sourcecode is available from sourceforge [SharkFW].

2 Building Distributed Applications with XTM, TMQL, TMAPI etc.

Topic maps comprise a whole family of formats and languages. Some of them are standardized. The ISO topic maps standard [TM] describes the data model and its semantics. It is the basis of the following formats and standards. XTM [XTM] is a XML schema for topic map representation. It is part of [TM]. The topic map query language [TMQL] is used to retrieve parts (fragments) of a topic map based on a query. Finally, TMAPI is an (not standardized) API for the management of topic maps. We have everything what's required to implement a topic maps application. The following figure illustrates the relationship between the components of a distributed TM application.

The application specific code is on top of the diagram. It uses TMAPI or TMQL to access and manipulate the underlying topic map(s). The topic map itself is stored in a component that is called *Knowledge Base*. There are no constraints on how topic maps are actually stored. The TMAPI and TMQL implementations hide KB specifics from TM applications and its developers.

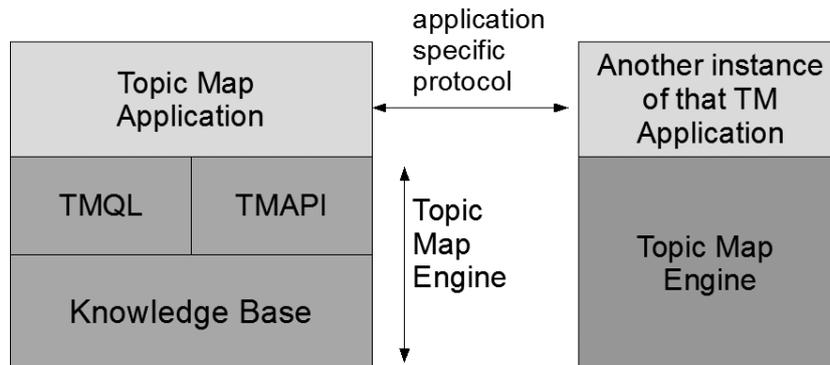


Fig. 1. Components of a Distributed **topic maps** Application

There is no explicit support for building distributed applications with topic maps. Application developers are free to use arbitrary network protocols to e.g. exchange topic maps with XTM or TMQL queries and their results. The communication issues are implemented in the application itself. Thus, it is an application specific protocol that enables communication between remote topic maps engines. TMSHare [TMSHare] is an example of such a distributed topic maps application.

3 P2P applications

P2P applications are a special class of distributed applications. There is no common definition of peers. In [Sc08] a model of autonomous context-aware peers is proposed – the ACP model. The basic ideas of the model are straightforward. Peers have the following features:

- A peer has its own knowledge base. There are no constraints on the knowledge representation formats used by the knowledge base. Of course, in this context a knowledge base can be assumed to be a topic maps engine.
- A peer observes its environment. Changes of the *environmental context* are recognized and can lead to an activity, e.g. the delivery of a message or the change of an internal status.
- A peer can send messages to other peers in its *environment*. The definition of environment remains very vague in the ACP model. It can be a local area network but also the WWW.
- Peers have (not necessarily unique) identifiers.
- Peers are autonomous. They can autonomously decide under which circumstances (based on the current environmental context, current connections to other peers, already exchanged messages, status of the knowledge base etc. pp.) messages are sent to other peers and what information are delivered.

Whenever peers take notice of each other they can decide to exchange messages and finally to exchange knowledge. Two processes have to be distinguished. Knowledge *extraction* is the process of taking knowledge from a peers knowledge base in order to send it to a remote peer. Knowledge *assimilation* is the process of retrieving knowledge and (partially) integrating it in the local knowledge base.

More formal, both processes can be defined as functions (in a Java like syntax):

```
Knowledge extraction(recipients, environmental context,  
status);  
  
void assimilation(sender, environmental context, status,  
knowledge);
```

Extraction generates a knowledge particle (in this context a topic map fragment). *Extraction* is influenced by the identity of the potential recipient, the current environmental context and the status of the peer. *Assimilation* integrates (parts of)

the retrieved knowledge. This process is influenced by the senders identity (if known), the current environment and the internal status.

There are, deliberately, no algorithms defined for any of the functions in ACP. This is up to an ACP implementation. The simplest implementation of *assimilation* is a topic map *merge*. The easiest implementation of *extraction* would be the usage of a static TMQL query. Both implementations would ignore the environment and the identity of the potential communication partners and would lead to a kind of distributed topic map but not to a network of autonomous context aware peers.

4 Shark framework – an implementation of ACP

The Shark framework [SharkFW] is an implementation of the ACP model. This supports the implementation of autonomous peers which can exchange knowledge in the described manner. The framework is written in Java and is currently available for J2SE and J2ME. The launch of version 1.0 is scheduled for september 2008. It is an extensible open framework with only a few requirements for underlying knowledge bases and communication environments used. A knowledge base must implement the functions *extract* and *assimilate*. An environment must allow the sending and retrieving of messages and should optionally be able to recognize changes (e.g. the appearance of peer). Version 1.0 comprises an UDP-Environment , a topic map with J2SE , a Bluetooth-Environment and a very simple knowledge base based on J2ME.

The main features of the Shark core are an API for autonomous peers and the implementation of a protocol engine supporting the P2P Knowledge Exchange Protocol (KEP).

5 Knowledge Exchange Protocol (KEP)

KEP is the Shark specific P2P protocol. There are four KEP commands. KEP was influenced by software agent protocols [KQML], [ACL]. In the following the four KEP commands will be briefly described.

- The *interest* command is submitted by a peer to indicate its *retrieving interest*. A peer can define what kind of information it is willing to receive . In Shark, this definition is simply done by naming a number of topics. Of course, XTM or LTM are preferred representation formats.

- The *offer* command is submitted by a peer to indicate its *sending interest*. A *sending interest* is the counterpart of a *retrieving interest*. A peer describes the kinds of information it is willing to send.
- The *accept* command is similar to the *interest* command but with slightly different semantic. *Accept* delivers a *retrieving interest*. The sender of an *accept* command expects to get a knowledge particle in reply (in return?).
- The *insert* command submits a knowledge particle, e.g. a XTM document. A sender will *extract* a fragment from its local knowledge base and send it to one or more recipients using a KEP *insert* command. The recipients will *assimilate* the retrieved knowledge. Both, *extraction* and *assimilation* algorithms are application (class) specific.

KEP is a stateless protocol. There is not even an implicit defined order of commands. Thus, KEP can easily be implemented with UDP, Bluetooth L2CAP and other datagram protocols.

Each KEP command contains the name of the sender (or *anonymous*) and the names of the potential recipients (or *anonymous*). There are some common usages of KEP sessions.

6 KEP scenarios – Internet peers

In the first scenario it is assumed that two peers with huge knowledge bases can establish a stable communication channel e.g. a TCP based connection in the fixed Internet. In the first step both peers can negotiate a *mutual interest*. This can be done by an exchange of interest/offer messages.

An example will explain the approach. Peer A has information about the latest music bands and movies. Peer B may be interested just in music. Thus, A would describe its *sending interest* with *music, movies*. (Note, this is an abbreviation. *Music* should be read as e.g. a topic standing for the concept of music. The string *music* can be a basename of this topic.). B would describe its *retrieving interest* with *music*. Now, A could send an *offer(music, movies)* command to B or B could send an *interest(music)* command to A.

If A receives (?) an *interest(music)* it can decide if and what to offer to B. In this example it would probably send an *offer(music)* to B. If B would receive (?) an *offer(music, movies)* from A it would learn that A has information on music and would probably reply with *interest(music)*. At the end of both sequences, B

knows that A offers music information. A knows that B is interested in music. Music is of *mutual interest*.

Now, B could send *accept(music)* to A. A would extract music information and send *insert(musicKnowledge)* back to B. Alternatively, A wouldn't wait for an *accept* and directly send an *insert* command.

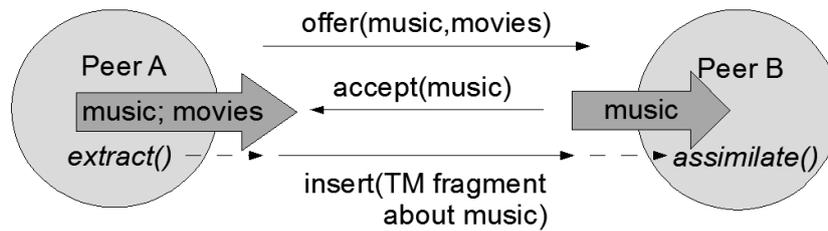


Fig. 2. Knowledge Exchange with Shark's Knowledge Ports

7 KEP scenario – Peers in spontaneous networks

Spontaneous networks are networks of mobile nodes which tend to enter and leave a network frequently. Moreover, a peer that was in a spontaneous network cannot be assumed to enter it again. A knowledge exchange strategy must be adopted to these characteristics.

A spontaneous network could e.g. be a network of two Bluetooth applications e.g. running on mobile phones. It takes several seconds to establish a connection. Bluetooth mobile phones can usually communicate within a radius of 10 m. Imagine two pedestrians (~ 3 km/hour and their mobile phones in their jackets) would pass each other. As soon as the distance is smaller than 10 m a spontaneous network could be established. The BT channel is of course dropped as soon as the distance is over 10m again. In this example, both mobile peers would have 12 seconds to establish a connection and to exchange information. Establishing a BT connection can already take about 10 seconds. Therefore there is no time left(?) for lengthy negotiations.

Another strategy should be used: Whenever a peer “sees” another peer in a mobile environment it should try to send relevant information as fast as possible. It could either send a (small) knowledge particle or an (retrieving or sending) interest with a different (e.g. IP or E-Mail) address for replies.

With the first strategy mobile peers would frequently get unsolicited insert commands. They would examine these knowledge particles and maybe assimilate parts of them.

With the second strategy mobile peers would just exchange their interests along with addresses to longer lasting peers i.e. internet peers. Such strategy is useful in environments which combine mobile and fixed peers.

8 Peer API / Knowledge Ports

The Shark framework supports the development of KEP based P2P systems. The following example code illustrates how a peer providing music information can be created.

```
Peer p = new Peer();

p.getKnowledgeBase().
addKnowledge("music","new album from madonna", "music news");

KP okp = p.createOKP("music");
okp.setVisible();
```

The first line creates a peer. New information is added to the knowledge base in the second line. Information consists of three parts: topic (“music”), creator (“music news”) and the information itself (“new album from madonna”). Note, this is also just an example and illustrates knowledge base access by Shark. If a topic map engine is used the code could be changed like this:

```
topic map tm = (topic map) p.getKnowledgeBase();
// do TM specific things, e.g. based on TMAPAPI or TMQL
```

The third line creates a knowledge port (KP). There are two knowledge port classes: incoming and outgoing knowledge port (IKP / OKP). An IKP is an object representing a set of information for assimilating information(?). An OKP is an object holding information for an extraction process. The function above is just a convenience function. The general KP constructor is defined as follows:

```
KP(KnowledgeBase kb, Peer peer, Context ctx, Context
interest, PeerName peers, boolean ikp, boolean okp)
```

The *kb* is the knowledge base which will extract or assimilate knowledge. The *peer* is the sending peer, *ctx* describes requirements for the environment, *interest* is either a *sending* or a *receiving interest* and *peers* describes the names of

potential communication partners of this port. Finally, two boolean values allow to define a knowledge port as IKP or OKP or both.

The convenience function in line 3 actually creates an OKP using the main knowledge base of the calling peer. The calling peer being the sender, defines no constraints on an environmental context. It defines just a single topic as *interest* (“*music*”) and allows to communicate with any peer that will be detected.

The last line makes this newly created OKP visible. Depending on the used environment, the KP will e.g. be published in a service directory and/or a broadcast is sent into the spontaneous network etc. pp.

Defining an IKP is as simple as defining an OKP:

```
Peer p = new Peer();
KP ikp = p.createIKP("music");
```

Both peers will be ready to exchange knowledge after both code fragments have been executed. A KEP protocol session will be performed whenever both peers can establish a communication channel. Knowledge will be exchanged if mutual interests can be negotiated .

As described above there are several KEP strategies. In version 1 just two are supported. Both have been described above. Default is the full negotiation. The KEP strategy of a knowledge port can be changed with the following command.

```
void kp.setStrategy();
```

9 Shark Engine

The Shark framework is an additional layer above a knowledge base, e.g. a topic maps engine and the application code. It provides a P2P protocol which is designed for loosely coupled systems, namely spontaneous networks but which also works on top of UDP or TCP in IP based networks.

The example above illustrated that e.g. four lines of code are sufficient to create a peer, enter sample data and to open a port for knowledge exchange. Just two lines of code are needed to define a peer that is interested in getting information about music. Shark hides the P2P communication protocol as well as the process of establishing a communication channel to other peers.

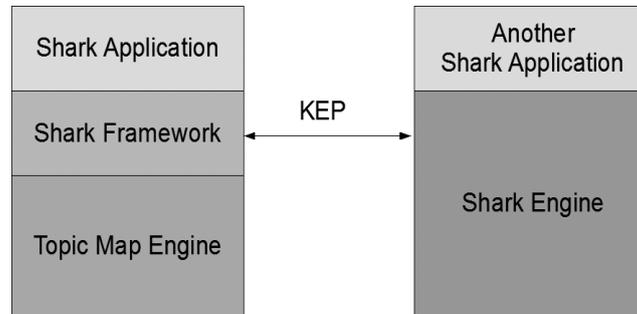


Fig. 3. Shark Application Components

The figure above also illustrates a feature of most frameworks for distributed systems. Shark framework already provides a protocol. The application code does not have to deal with protocol specific issues. It only has to define rules for knowledge exchange.

Application independent protocols are an advantage in general: an application specific protocol might potentially change whenever the application is changed. A Shark application isn't even able to change the KEP protocol at all. It can only handle received knowledge or interests in different ways.

10 Shark peers versus software agents

Shark peers have something in common with software agents which are used in the field of distributed artificial intelligence. Nevertheless, there are major differences: Software agents are meant to be entities that can fully replace human users in a dedicated application domain. Agents can simulate plans, strategies of human users as well as (in a reduced and limited manner) feelings and biases.

Shark peers are just containers holding information and algorithms for knowledge exchange. Shark peers are parametrized and run on behalf of human users but they would and could never be seen as a *substitute* of a human user. From a very abstract perspective Shark peers can be compared to an intelligent filtering system but not to a replacement of personal strategies.

11 Shark peers versus distributed systems

A P2P system is a distributed system. A system based on Shark is a distributed system as well. The concept of autonomy makes it different from e.g. file exchange systems and music exchange platforms. A Shark peer decides (based on its algorithms) if and what kind of information shall be exchanged. In other P2P systems users browse through a collection of information and decide i.e. what to download. Peers are passive entities that make their local information bases accessible to remote peers.

There are distributed systems that hide distribution. Distributed databases combine several databases and present them as a single virtual database to software developers and users. The distribution is hidden. Middleware systems like CORBA, EJB etc. also hide distribution. Shark doesn't. Developers and users are aware of the fact of distribution. Thus, Shark can and should only be used for applications which are not meant to hide the fact of distribution.

12 Summary and outlook

The Shark framework is an implementation of the model of autonomous context aware peers. It is an open framework. The Shark core has just very weak assumptions on knowledge base features. The Shark protocol KEP is stateless and can easily be implemented on top of datagram protocols like UDP and Bluetooth L2CAP. Currently, Shark is implemented in Java (J2SE and J2ME). Nevertheless, Shark is far from being finished. Even version 1.0 can only be seen as a very first step.

Mobile P2P systems should support a broad range of hard- and software. In the next steps Shark will be ported to Google's Android and to Apple's iPhone. Furthermore, applications are needed to proof the concept and to give input for further revisions of the framework. This paper shall also be understood as a call for participation. Shark is published under the LGPL on sourceforge. Shark is an acronym. It stands for Shared Knowledge. Let's share Shark!

References

- [TMAPI] *topic map API*: <http://www.tmapl.org/>
- [TMQL] *ISO/IEC 18048: topic map Query Language*.
<http://www.isotopicmaps.org/tmq1/>

- [KQML] Finin, T., Weber J., Wiederhold, G., Genesereth, M. Fritzon,R., McKay, D., McGuire,J., Pelavin, R., Shapiro, S., Beck, C.: *Specification of the KQML Agent-Communication Language - plus example agent policies and Architectures*; June 1993
- [TMShare] Ahmed, Kal: *TMShare – topic map Fragment Exchange in Peer-to-Peer-Application*. In: Proceedings of XML Europe 2003, London 2003.
- [ACL] FIPA Communicative Act Library Specification / Foundation for intelligent physical agents. 2002 – FIPA standard
- [Sc08] Schwotzer, T.: *Ein P2P system basierend auf topic maps zur Unterstützung von Wissensflüssen*; Vdm Verlag Dr. Müller, April 2008, ISBN 978-3639008371
- [SharkFW] *Shark framework – Shared Knowledge framework*;
Sourceforge: <http://sourceforge.net/projects/sharkfw/>
- [SG02] Schwotzer, T., Geihs, K. 2002. Shark - a System for Management, Synchronization and Exchange of Knowledge in Mobile User Groups. In Proceedings of the 2nd International Conference on Knowledge Management (I-KNOW '02), 149-156. Graz, Austria
- [MS05] Maicher, Lutz; Schwotzer, Thomas.: *Distributed Knowledge Management in the Absence of Shared Vocabularies*; In: *Proceedings of the 5th International Conference on Knowledge Management (I-KNOW'05)*. Graz / Austria, July 2005
- [TM] ISO/IEC 13250: *topic maps*; December 1999
- [WfMC] The Workflow Management Coalition:
<http://www.wfmc.org/>
- [XTM] Pepper, S., Moore G.: *XML topic maps (XTM) 1.0*; March 2001

Towards an automatic semantic integration of information

Jörg Wurzer¹ and Stefan Smolnik²

¹ IQser AG, Chlupfgasse 2, 8303 Bassersdorf, Switzerland
joerg.wurzer@iqser.net, <http://www.iqser.net>

² Institute of Research on Information Systems (IRIS), European Business School (EBS)
Rheingastr. 1, 65375 Oestrich-Winkel, Germany
stefan.smolnik@ebs.edu, <http://www.ebs.edu/iris>

Abstract. With their expanding information assets and the increasing importance of the knowledge factor, organizations are increasingly challenged to efficiently support knowledge management processes with appropriate integration and retrieval technologies. Besides traditional information retrieval approaches, the use of semantic technologies like Topic Maps is also becoming more important. This paper proposes a technology framework for the automatic semantic integration of information. Based on various information repositories, topics and topic associations are created automatically in real time. In addition, the first results from a proof of concept in conjunction with the European company EADS provide further insights into the proposed framework's applicability in practice.

Key words: Semantic information integration, information retrieval, automatic approach, Topic Maps, proof of concept

1 Introduction and motivation

In today's office environments, work is affected by the escalating ratio of electronic information, which – mostly in the form of documents – is increasingly badly structured. The challenge is therefore no longer to obtain information, but rather to use it effectively and efficiently by specifically identifying relevant information from the large mass of information available.

Another substantial trend is that the resource knowledge and its management are gaining in respect of organizational meaning (see e.g., [VIN00, p. 3], [Rie04, p. 88f]). In the past decade, knowledge has become a crucial competitive factor as it has a large potential to generate value. Those organizations that are able to utilize their knowledge resource effectively and efficiently can thus obtain a sustainable competitive advantage from such value generation (see [Smo06, p. 3f]).

Information technologies – particularly technologies relating to information retrieval and structuring – address the challenges arising from information overflow and extensively support knowledge management to realize sustainable competitive advantages (see e.g., [Hec02, p. 2]). In the course of the World Wide Web's development into the Semantic Web, increasing attention has been paid to concepts and technologies based on semantic approaches. Besides the creation of the Semantic Web, semantic technologies like Topic Maps can also be used to develop intra-organizational information systems that address the challenges described above.

In market-driven systems, there are essentially two options for finding and retrieving information. These are either a full-text search or accessing a hierarchy of directories. Both approaches have indeed proven themselves, but also have disadvantages (see e.g., [Smo08]).

The advantage of full-text search is its operational simplicity, which has proven itself with, for example, Google. The user just enters a keyword or a combination of keywords and receives a result. However, there are also some disadvantages:

- The quality of the result is dependent on the selection and combination of keywords used in the search. If the keyword is a commonly used word, the search result can be quite large. The result is presented in the form of a long list of entries, which the user has to evaluate individually. This process is rather time-consuming.
- Full-text search only considers the occurrence of one or more keywords in a single document. It does not consider the word's meaning in the overall context of the document. It may also describe a completely unimportant issue to the overall document.
- The relevance of a search result is another major challenge for search engines. They do not consider the intention of the searcher or the context of his information need. Consequently, search engine providers like Google introduced the idea of sorting search results, which evaluates results in the form of how often a web page is referred to by other websites.

The advantage of directory hierarchies is that they can be created manually by aligning the logic of the meaning of the contents and their applicability. This

requires a well-conceived structure that takes the user's interests into account. However, this also has disadvantages:

- The information containers (whether documents, emails, or web pages in a portal) are collated in a specific way. In practice, there are often various approaches to accessing an information repository. For example, sometimes the customer view of the information is interesting, then the product view, and finally, a personally customized view. However, a directory has to establish a structure. Modifications, which have resulted over the life of the organization or the definitions of the project, are difficult to implement.
- Most information is important in more than one context. However, multiple relationships between information containers in a directory tree generally lead to redundantly stored information. This is not only a question of resources, but also an obstruction to the modification of any documents.
- The quality of the directory tree depends on the author. The directory tree reflects the current state of knowledge and of the organization. There may be information containers that can not be integrated into this directory tree, for which a work-around has to be found by means of "miscellaneous files" or similar solutions.
- The information containers in a directory are normally limited to files or objects of a specific type, such as email. Most systems do not possess a sufficiently fine granularity with regard to the information containers and their relationships. This has a huge disadvantage for the overview and organization of information.

2 Making each piece of information accessible through various contexts

The proposed technology framework offers an alternative to these two approaches. Information containers are cross-linked according to content-based criteria, which allow a topic map of information containers to be created. This topic map can be found by means of its context based upon the current focus of the user interest. Given that the user is interested in the business dealings, contracts, products, employees, and service calls of a specific customer, all of this information should be associated with that customer. The information must come from various systems, and should not be fed into a central relational database, in which the linkages would be unalterable. Any possible context is

conceivable: A person, a project, a very special document, a meeting, a job, a message, and much more. In each case, the proposed technology framework displays the information containers that are important to that context. A profile of a person may refer to publications within and out-side a company, to projects in which that person participated, or to contacts. A document can refer to the author, to other articles enlarging the topic, or to topic areas that are, for example, defined as part of a company's research and development.

This results in a new procedure for the retrieval and interpretation of information containers. Instead of searching for terms, or following a directory path, the search focuses on the context that is currently of interest. Naturally, it is also possible to combine contexts to filter information. The user will then immediately have information that is meaningful to a specific context. This allows the user can discover connections in the data inventory and draw conclusions that may result in new knowledge. Users may, for example, recognize that a certain employee has specific skills, which distinguish him in his project experiences, publications, and personal contacts, as well as training.

By utilizing the proposed technology framework, contexts are no longer abstract concepts, but rather concrete information containers. This framework makes it possible to navigate from information container to information container throughout the data inventory by means of automatically generated topic associations. Unlike a directory system, the topic map is not hierarchical and the topic associations always lead to additional information containers. This is very helpful for the exploration of data, as well as for targeted searches. The user can move from one person to his publications and from the publications to sites on the Internet that have already taken up the subject or represent an alternative point of view.

This is a new paradigm for accessing and organizing information. All of the information from a variety of systems becomes integrated and granularly classified within a context. In a company, enterprise resource planning (ERP) and customer relationship management (CRM) systems may consequently be as connected as document management systems, communication servers, and collaborative solutions.

3 Proof of concept for EADS

The European company EADS is organized into five divisions. The business unit "Defense and Communication Systems" is part of the "Defense and Security Systems" division and forms also the systems house of the EADS group. The

subsequently presented proof of concept has been conducted jointly with this business unit.

The requirements of a proof of concept test for EADS was an analysis of unstructured data on military information. The unstructured data were delivered as Microsoft Word, PDF and Excel documents. The proposed technology framework had to automatically create a hierarchical taxonomy and relations between the documents. The relations had to be comprehensible. The user had to have the opportunity to see why a relation had been established. The result of the analysis was compared with manually created ontologies.

Figure 1 shows the user interface of this proof of concept test. As the data referred to secret military information, this screenshot shows data from another source. On the left there is a category tree. Although this tree has been automatically created, there are only a few top terms. The user can choose a path in this tree to select the data shown on the right site. The weight indicates the extent to which a document relates to the selected path in the tree. If the user selects one document, all the related documents are shown in the content pane on the right under the result list. Here, the weight indicates the extent to which a document relates to the selected document. The “Details” button leads to a pop-up window explaining the link between the two information objects, which can, for example, be text similarity, or specific attributes (see the following section for a description of the algorithms). The user selects a document in this content pane and the system shows related objects. This offers an opportunity to navigate through the entire document base by means of topic associations.

Filename	Filetype	Filesize	Last Modified	Weight
MECKL.HTM	HTML Document	6098 bytes	1996-09-15 12:50:30	1.0
HAMBURG.HTM	HTML Document	11643 bytes	1996-09-15 12:50:18	0.9674754738807678
SAARL.HTM	HTML Document	8199 bytes	1996-09-15 12:51:10	0.6552774906158447
BERLIN.HTM	HTML Document	30332 bytes	1996-09-15 12:50:02	0.6435524353981018
SCHLESWH.HTM	HTML Document	2456 bytes	1996-09-15 12:51:36	0.6360944509506226
SACHSEN.HTM	HTML Document	26620 bytes	1996-09-15 12:51:32	0.6027675867080688
HESSEN.HTM	HTML Document	17512 bytes	1996-09-15 12:50:26	0.6010527610778809
RHEINLPF.HTM	HTML Document	8720 bytes	1996-09-15 12:51:04	0.595011830329895
NIEDERS.HTM	HTML Document	16023 bytes	1996-09-15 12:50:38	0.5791889429092407
SACHSANH.HTM	HTML Document	16179 bytes	1996-09-15 12:51:18	0.5791889429092407
BAWUE.HTM	HTML Document	24583 bytes	1996-09-15 12:49:38	0.574716329574585
BAVERN.HTM	HTML Document	25080 bytes	1996-09-15 12:49:50	0.574716329574585
BRANDEN.HTM	HTML Document	15407 bytes	1996-09-15 12:50:08	0.5679414868354797
MDW.HTM	HTML Document	69114 bytes	1996-09-15 12:51:00	0.5664574192708623
Filename	Filetype	Filesize	Last Modified	Weight
WAND.HTM	HTML Document	219 bytes	1996-08-25 01:05:38	0.136082768440264658
OFFEN.HTM	HTML Document	1212 bytes	1996-09-27 06:29:14	0.011930519714951515
ANL.HTM	HTML Document	828 bytes	1996-08-27 10:10:04	0.009699944406747818
bbbbaum.htm	HTML Document	829 bytes	1996-08-27 07:22:56	0.009699944406747818
bbelfe.htm	HTML Document	828 bytes	1996-08-27 07:27:18	0.009699944406747818
BILDGR.HTM	HTML Document	828 bytes	1996-08-27 07:17:10	0.009699944406747818
HAND.HTM	HTML Document	826 bytes	1996-08-27 07:36:32	0.009699944406747818
INANDV.HTM	HTML Document	828 bytes	1996-08-27 07:37:56	0.009699944406747818

Fig. 2. User interface of the proof of concept

The tree of terms gives the user an overview of the content of the document base as well as related fundamental facts. In this case, the tree shows that “Biber” is a tank. The quality of information in the tree, as well in the topic map of document is high and can compete with that of a manually created ontology.

4 The uniform information layer

In order to build this topic map, all of the information containers from various systems are connected first. This is conducted by the Uniform Information Layer. With the help of a plug-in framework, an interface is implemented for each type of information container, which translates the respective container into an object with a format consistent with the technology framework’s core server. It does not matter if the information container is an unstructured file or a structured record from a database or application. The implementation of the interface is very easy. The software engineer just has to take care to get content objects from the sources and transform them into a generic content object. This generic content object is semantically typed and contains a list of attributes and meta-data, which are used by the core engine. Distributed information is consolidated by the Uniform Information Layer into a huge data pool, without redundantly storing data.

The result is that a comprehensive full-text search is just as possible as a complex search using specific information container attributes. Thereby, the context for data access can be retrieved in a selective manner as indicated above. The topic map is the result of a three-step analysis procedure. The topic map is continuously adjusting to new circumstances as soon as a new information container is created, modified, or deleted. This happens in real time, without the need to reprocess the entire data inventory. This is necessary for huge data inventories as well as processes. They can be smoothly connected with the core server, which possesses an event-driven architecture. Individual process steps can be monitored and controlled. Jobs that are created by this are automatically provided with all of the information necessary to complete the task.

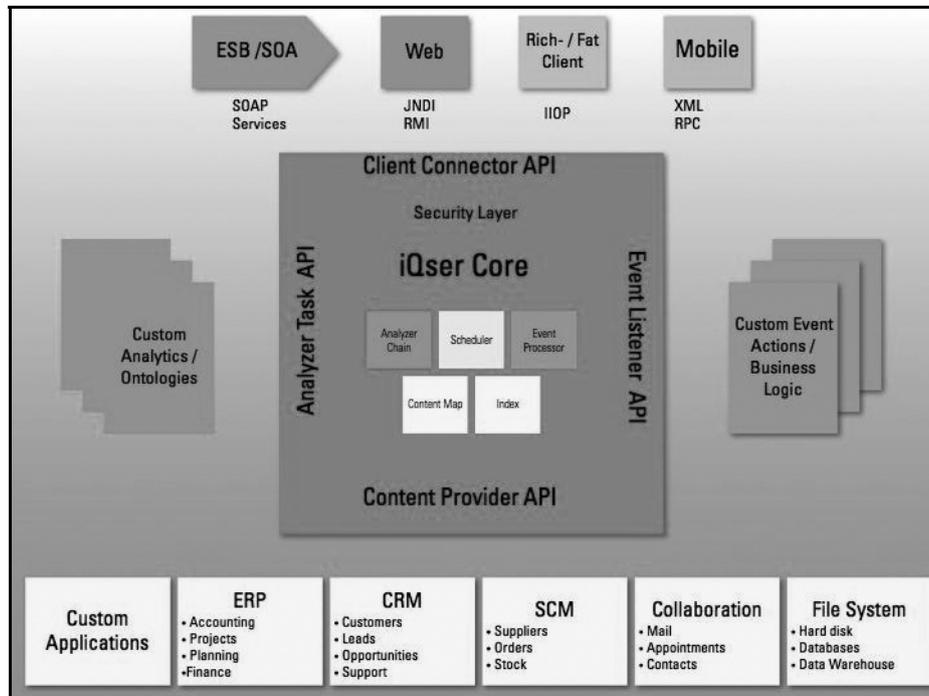


Fig. 3. Technology framework

The architecture of the core server consists of the main functionalities (see figure 2), which are integrated in the semantic middleware. The server recognizes new or changed content, and starts indexing and analyzing this content. The result is stored in both an index and a content map representing a topic map. Within this content map, pairs of content objects can be linked by n relations depending on

the reasons of the link's generation. Furthermore, each link is weighted according to its relevancy. A security layer filters or protects data by reflecting the roles and security requirements of an organization.

The semantic middleware provides four interfaces (see figure 2): The Content Provider API integrates heterogeneous data sources like described above. The Analyzer Task API provides functionalities to integrate additional analyses. An organization may have special needs to create relations or rules like in the form of an ontology. The Event Listener API integrates a business process engine to use new or changed data as a trigger or a control instance for processes. Eventually, the Client Connector API provides different concepts for representing generated information in a user interface. The user interface could be a rich client or a web-based application. Furthermore, the semantic middleware could be integrated in the SOA environment of an organization.

The core server is implemented in Java and is therefore executed on an application server following the J2EE specification. Thus, the server is capable of processing terrabytes of data. We conducted several performance tests. One of these tests included 3 GB of data to be indexed and analyzed. The document formats were MS Word, plain text, MS Excel, MS PowerPoint, and HTML. The initial indexing and analysis were completed within 14 hours. The system was constantly fully loaded. More than 70% from overall CPU resources were spent for I/O waits. The CPU memory needed less than 400 MB memory. The following test system was used:

- Hardware: Pentium(R) Dual Core 3 GHz, 2 GB RAM
- Software: Windows XP 2002 SP3, JBoss 4.0.4 GA, Sun JDK 1.5_12
- JBoss JVM heap size configuration: -Xms128m -Xmx512m

5 A combination of three analysis procedures

The analysis performed by the core server combines three procedures, which together deliver an optimized result. The analysis process is controlled by a journal, which contains every modification in the data inventory as well as all user interactions with the system.

1. *Syntactic analysis*: The core server understands this to be an association of information containers based on key attributes. The key attributes are defined by the plug-ins indicated above. This may be the sender and recipient in the case of an email, for example. The core server searches for a correlation to this

attribute in the huge data inventory. This is one way of naming an email address in a person's profile or a document. The email correspondence is bound to a person in the result and is retrievable without manual assignment. Every type of information container can be given any number of key attributes. Normally, these are structured elements, such as a customer number or a project reference ID.

2. *Pattern analysis*: The core server understands this to be the creation of a pattern comparison between various contents. For this, the analysis procedure extracts those terms, which are meaningful, from a text and transforms this mass of words into a data query, whose result is listed similar to an information container. Each information container is provided with a match value between 0 and 1. Full matches receive the value < 1 , while the value 0 is a threshold value indicating complete difference. The value 1 is provided for creating topic associations manually. The meaningful terms are processed into an information container based upon the reciprocal word frequency and selected. Words with a lower frequency have a higher significance. The terms with the highest significance are incorporated into the group of meaningful terms. In comparison, the match value is calculated based upon the number and frequency of conforming words. Even the word spacing plays its part.
3. *Semantic analysis*: This procedure is based upon a perception of the philosophy of languages: that the meaning of language arises from its usage and is continuously redefined in new ways. The meaning of information containers arises analogously through their usage. They change over the course of time and shift the context or relevancy of their usage until obsolete documents become useless. For this reason, the core server monitors all of the user interaction from the creation of information to modification and even deletion. Even simple retrieval is monitored. A special algorithm reprocesses the relationship of information containers after each journal entry. If two information containers are often retrieved or edited in a given sequence, the system can make the conclusion that a meaningful context exists. This grows as the pattern is repeated. The relevancy of an topic association may however decrease, if one set of topic associations is used more often than another. In this manner, every topic association receives an assigned relevancy, which enables the user to make selections more easily when, for example, several documents were assigned to one person.

It does happen that two information containers are associated with each other based upon two or even all three of the procedures. As a fourth option, manual

association should also be mentioned. It may good be that there are several reasons why two information containers have been connected with each other by syntactic analysis. For this reason, there may be not just one topic association between two information containers, but rather n topic associations. The weighting of the topic associations, which are only displayed as one, is taken from an average of the individual weightings. The reason why a topic association has been created can be optionally shown to the user. In this manner, the system remains comprehensible.

6 Querying associated information

Information containers can be reached through their context. The technology framework offers to use a query for this, which specifically selects the data inventory. Unlike a pure full-text search engine, the core server can search for the specific attributes and attribute values of an information container. Bookmarks assist with remembering complicated queries and can always be retrieved again. Now, it is possible that there are many additional information containers associated with a single information container, as with a project. The context search becomes important in this event. For this context search, a query is applied to the information container, which is associated with the specific information container.

Every modification in the data inventory triggers an event, which can be used in the core server as the cause for an action. This even includes the discovery of a new context. An action might be the notification of a user or also the fact that the system is modifying or creating new data. In this manner, a service inquiry by a customer can be automatically answered, be forwarded to the corresponding clerk, and a ticket for the reply to the inquiry created. Complex processes can even be connected to, and controlled by, existing deployed business process management (BPM) solutions. The potential applications are innumerable. In addition to process control across systems in a company, research, competition and opinion trends can be monitored. All of the information is available as soon as it is published, whether in a company network or in the Internet. The starting point for the monitoring does not have to be one or more search terms; it might be complete documents describing a subject. This last item can be the starting point for a search and pull in relevant information from a huge pool, just like a magnet. The description of research topics, products, or markets would be conceivable. Even patent specifications can be compared with the current state of research.

7 Conclusions and areas for future research

In this paper, we have presented a technology framework for the automatic semantic integration of information. Besides introducing basic paradigms and concepts, we have also indicated a combination of three analysis procedures. In addition, some first results from a proof of concept in the European company EADS have been presented. In the context of this case, an analysis of unstructured data on military information as well as an automatically created hierarchical taxonomy and respective relations between information objects have been specifically investigated. It could be demonstrated that the presented automatic approach could keep up with the intellectual modeling of knowledge structures.

One major area for future research is gaining more detailed insights and practical experiences by applying the proposed framework in various environments. Besides the conceptual considerations covered by this study, the described concepts and benefits make it necessary to determine practical impacts when the framework is employed, i.e. during more proof of concept projects in real life environments. In addition, business cases should be defined for deploying and running the framework to gain further economic insights.

References

- [Rie04] Riempp, G.: Integrierte Wissensmanagement-Systeme – Architektur und praktische Anwendung. Springer, Berlin, London etc., 2004.
- [Smo06] Smolnik, S.: Wissensmanagement mit Topic Maps in kollaborativen Umgebungen – Identifikation, Explikation und Visualisierung von semantischen Netzwerken in organisationalen Gedächtnissen. Shaker, Aachen, 2006.
- [Smo08] Smolnik, S.: Convergence of classical and semantic search – evidences from a chemical case, in: Lecture Notes in Computer Science, Volume 4999, 2008, pp. 14-24.
- [VIN00] Von Krogh, G.; Ichijo, K.; Nonaka, I.: Enabling Knowledge Creation – How to Unlock the Mystery of Tacit Knowledge and Release the Power of Innovation. Oxford University Press, New York, USA, 2000.

*Towards a new generation of
Topic Maps engines*

Towards a second generation Topic Maps engine

Xuân Baldauf¹ and Robert Amor²

¹ University of Auckland, New Zealand
xuan--tm4j2--2008--tmra.de@academia.baldauf.org

² Department of Computer Science, University of Auckland, Private Bag 92019,
Auckland, New Zealand
trebor@cs.auckland.ac.nz

Abstract. The core of the second generation Topic Maps standards (TMDM, XTM2.0) has been finalized, yet the uptake is still slow. In this paper, we highlight engineering considerations for a novel backend for the TM4J open source topic maps engine, which is currently in development, but already usable for some purposes. As the name suggests, the “TMDM” backend is designed to reflect the TMDM specification closely. In fact, it is much closer to the TMDM than to the internal legacy TM4J data model (which is based on the XTM 1.0 data model). This motivates a bridging layer between the TMDM and the XTM 1.0 data model. We emphasize how merging is implemented in the “TMDM” backend and conclude with some synthetic merging benchmarks of the current “TMDM” backend prototype.

Keywords: TM4J, TMDM, Topic Maps engine, Merging, Instant Merging, Dynamic Merging

1 Introduction

A new generation of Topic Maps standards (the Topic Maps Data Model [ISO13250-2], XTM 2.0 [ISO13250-3]) was finalized in 2006, yet adoption in the community remains slow. TM4J¹ is an open source Topic Maps engine written in Java, mainly by KAL AHMED. The most recent release (TM4J 0.9.7, published in 2004) is based on the older XTM 1.0 [XTM1.0] standard. While development activity on TM4J slowed after 2004, TM4J is still the most comprehensive open source Java Topic Maps Engine, and several projects build on TM4J. Thus, TM4J clearly needs an update to support the new Topic Maps standards. Updating TM4J is preferable to designing a completely new and

¹ See <http://tm4j.org/>

independent Topic Maps engine, as, in the case of an ideal update, all TM4J legacy application can build on an updated TM4J without need for modification on their part. This in turn leverages the existing TM4J applications and allows them a smooth migration path to the new Topic Maps standards.

In this paper, we show the design principles of a novel backend for TM4J, which is anticipated to lead TM4J to version 2.0. We will use the term “TM4J1” for the branch of TM4J which keeps the architecture of TM4J 0.9.7 and will, in particular, not support the TMDM. We will use the term “TM4J2” for the branch of TM4J which undergoes the major architectural changes we are describing here.

1.1 Assessment

When starting to work with TM4J1, we were in need of a topic maps engine which would be able to consume many small automatically generated XTM 2.0 files and merge them into a large XTM file. However, TM4J1 supported neither the syntax of XTM 2.0 nor the semantics of XTM 2.0 (which is specified by the TMDM). Internally, merging is only done on request, not instantly, as the TMDM mandates in section 6.1: “*Any change to a topic map [...] shall be followed by [...] merging*”. Each such merging request would apply to the whole topic map, making “simulated instant merging” (by requesting such merging after every small change) infeasible with respect to performance. Furthermore, the TM4J1 API is outdated in multiple ways. First, the TM4J1 API is based on an older version of the Java language (e.g. it lacks support for Java Generics). Second, the TM4J1 API is slightly, but still significantly incompatible with the TMAPI [TMAPI1.0SP1], giving rise to a need for a wrapping-layer around TM4J1 objects to make them appear as TMAPI objects. The TMAPI itself (as of version 1.0) has not yet been updated to the TMDM, thus the names of the classes and methods in the TMAPI 1.0 do not exactly match the names of classes and properties of the TMDM.

For these and other reasons, the following desired features have been identified:

1. Internal support for the TMDM,
2. Support for XTM 2.0,
3. Instant merging,
4. Dynamic merging (where the individual components can still be identified),
5. Support for modern Java language features, such as Java Generics,

6. Let TMDM guide the naming of classes, methods and fields,²
7. Translation between the TM4J2 data model and the TM4J1 data model³.

2 The TMDM backend

The novel “TMDM” backend for TM4J is designed upon the well known principle of separation of concerns. This principle guides

1. that storage of Topic Maps (in RAM) should be separated from a merged view of Topic Maps,
2. that a TM4J1 data model view should be separated from a TM4J2 data model view,
3. that interfaces should be separated from implementations,
4. that interfaces themselves should be separated by concerns,
5. that event handling should be separated.

This is why there is not only one set of classes (or interfaces), but five:

1. The interfaces for TMDM data with read-write access.
2. The interfaces for TMDM data with read-only access.
3. The classes to store TMDM data.
4. The classes to view TMDM data in merged form.
5. The classes to access TMDM data through the TM4J1 data model.

Additionally, a new handling system to efficiently communicate events between the different layers of objects has been devised.

In the following sections, these layers and subsystems are described. Figure 1 (below) provides an overview over all these layers.

2.1 TMDM interfaces layer (read-write access)

This layer contains interfaces for representing TMDM objects, all within the package `org.tm4j.topicmap.tmdm`:

² If the TMDM guides the naming of classes, methods and fields for the “TMDM” backend as well as for the upcoming TMAPI 2 standard, then the “TMDM” backend may be automatically compatible with the upcoming TMAPI 2 standard, making a separate wrapping layer (as in TM4J1) unnecessary.

³ When supporting the TMDM but, at the same time, serving as a backend for TM4J1 applications, there is a need for translating between the TM4J1 data model (which is the data model of XTM 1.0) and the TM4J2 data model (which is the TMDM).

- | | |
|-----------------------|---------------------------|
| 1. TopicMap | extends Reifiable |
| 2. Topic | extends TopicMapConstruct |
| 3. TopicName | extends Scopeable |
| 4. Variant | extends Scopeable |
| 5. Occurrence | extends Scopeable |
| 6. Association | extends Scopeable |
| 7. AssociationRole | extends Reifiable |
| 8. Scopeable | extends Reifiable |
| 9. Scope | |
| 10. Reifiable | extends TopicMapConstruct |
| 11. TopicMapConstruct | |

Each of these interfaces contains methods to read and write properties of the TMDM item type they represent. For example, the `Topic` interface contains, among others, the following declarations:

```
public boolean addSubjectIdentifier(Locator subjectIdentifier);
public boolean removeSubjectIdentifier(Locator subjectIdentifier);
public Set<Locator> getSubjectIdentifiers();
```

As another example, the `TopicName` interface contains, among others, the following declarations:

```
public void setType(Topic type);
public Topic getType();
public void setValue(String value);
public String getValue();
```

Scopeable and Scope. The interface hierarchy here differs from the TMDM class hierarchy in that the interfaces `Scopeable` and `Scope` are introduced. While in the TMDM specification, *scope* is defined verbally (“*All statements have a scope.*”), a reflection of this definition is lacking in the original TMDM class hierarchy: *scope* is left to remain an arbitrary set of topics in each statement without a unique identity. This is changed in TM4J2. The rationale behind this is the reasonable assumption that the set of distinct scopes in a typical topic map is much smaller than the set of scopeables (that is, the set of statements). If this assumption is true, then instead of storing a mutable set of topics for each `Scopeable` (which typically consumes at least a Java array object header and pointers to each of the topic objects), it is more memory-efficient to just store a mutable pointer to an immutable `Scope` object. It is also assumed that, at query time, this compression increases cache-locality, as the number of distinct scope objects (`Scope` objects vs. sets of topics) to be traversed is much smaller. Furthermore, in case two topics of the same scope merge, changing the affected

Scope object⁴ is much cheaper than changing, or even just keeping track of, all affected Scopeable objects. However, as TM4J2 is not fully implemented yet, and also because there is, to date, no well-agreed Topic Maps benchmark suite (consisting of demo topic-maps in various serialization formats and demo queries in the yet to be finalized Topic Maps Query Language), all these considerations are merely theoretical and are still in need of performance evaluation.

2.2 TMDM interfaces layer (read-only access)

This layer contains interfaces for representing TMDM objects which are only to be read, but not to be written, all within the package `org.tm4j.topicmap.tmdm`:

1. `ReadableTopicMap`
2. `ReadableTopic`
3. `ReadableTopicName`
4. `ReadableVariant`
5. `ReadableOccurrence`
6. `ReadableAssociation`
7. `ReadableAssociationRole`
8. `ReadableScopeable`
9. `ReadableScope`
10. `ReadableReifiable`
11. `ReadableTopicMapConstruct`

Each of these interfaces contains methods to just read properties of the TMDM item type they represent. They are stripped-down versions of their read-write counterparts. For example, the `ReadableTopic` interface contains, among others, the following declarations:

```
public Set<Locator> getSubjectIdentifiers();
```

As another example, the `ReadableTopicName` interface contains, among others, the following declarations:

```
public ReadableTopic getType();
public String          getValue();
```

⁴ Giving up immutability of Scope objects leaves opportunity for two Scope objects being equal. While avoiding this repetition is the very reason to have Scope objects in the first place, actually having such repetition just in rare cases has only a tiny effect on the ratio between actual memory savings and possible memory savings by this method.

The rationale for having a layer of TMDM interfaces which just allow read-only access, separate from TMDM interfaces which allow read-write access, is the case of Virtual Topic Maps. A Virtual Topic Map⁵ is a view⁶ on something which looks like a topic map, but may not actually be a (modifiable) topic map itself. As one objective of topic maps is to be able to represent the structure of almost any type of information, it is only consequential to reformulate about almost any information source⁷ as a topic map. However, changing such a topic map is often (unless it is materialized) not possible directly; however, changing the information source, and having this change reflected in the topic map view, is possible. If the translation between the information source and the topic map view is a one-way-process (i.e. only from the source to the topic map view and not the other way around) for theoretical or practical reasons, then there is no sensible way of implementing the setter methods which modify topic maps. If, on the caller side, only getter methods are needed (for example, if a GUI view or another topic maps view is built on the topic maps view), then the more adequate interface between these two sides is the set of TMDM interfaces which just allow read-only access.

Each read-write TMDM interface extends the corresponding read-only TMDM interface. Note that e.g. the return type of `ReadableTopicName.getType()` is not `Topic` but `ReadableTopic`, while the return type of `TopicName.getType()` is `Topic`. The reason is that the read-only TMDM interfaces have to be closed within themselves, i.e. they should not point into the world of read-write TMDM interfaces. Note also that narrowing the return type when overriding (from `ReadableTopic` to `Topic`) is a feature of Java 1.5, thus unavailable at the times the original TM4J1 architecture was designed.

⁵ It is unclear to whom to trace the term “Virtual Topic Maps”. However, the earliest instance of explaining this term, which we could find, is following mailing-list post of STEVE PEPPER:

<http://www.infoloom.com/pipermail/topicmapmail/2001q3/003190.html>

⁶ A view is something which depends on, and its contents are defined by, what is viewed.

⁷ “Any information source” does not preclude topic maps themselves as information sources. For example, as ROBERT BARTA points out in his talks about TMQL, it may be perfectly reasonable that a topic map is an information source for an inference engine which takes that topic map as input, infers new facts from existing facts, and exports a topic map view as output. Note that the topic map constructs of the output may be generated on demand, i.e. only when a query is active. This way, the memory requirements for such an inference engine can be much smaller than the memory requirements if the exported topic map view was materialized.

2.3 TMDM Basic implementation layer (read-write access)

This layer contains classes for representing TMDM objects, all within the package `org.tm4j.topicmap.tmdm.basic`:

1. `BasicTopicMap`
2. `BasicTopic`
3. `BasicTopicName`
4. `BasicVariant`
5. `BasicOccurrence`
6. `BasicAssociation`
7. `BasicAssociationRole`
8. `BasicScopeable` (abstract class)
9. `BasicScope`
10. `BasicReifiable` (abstract class)
11. `BasicTopicMapConstruct` (abstract class)

Each of these classes implements the appropriate read-write TMDM interface.

In a Model-View-Controller design, this layer contains the model. That means that all actions to modify a topic map are actions on objects in the Basic layer, the objects in the Basic layer act as mere storage. Thus, questions about whether two `BasicTopicMapConstructs` are to be merged, or not, are not answered here. For example, even if two `BasicTopic` objects are to be merged, it is not possible to query the merged set of the merged topic's `BasicTopicNames` (directly) if only a reference to one of these `BasicTopic` objects is available. Effectively, the Basic layer represents topic maps as if the merging rules did not exist. However, actions on `BasicTopicMapConstructs` induce events, which are typically forwarded to the Merged layer.

2.4 TMDM Merged implementation layer (read-only access)

This layer contains classes for representing TMDM objects, all within the package `org.tm4j.topicmap.tmdm.merged`:

1. `MergedTopicMap`
2. `MergedTopic`
3. `MergedTopicName`
4. `MergedVariant` (currently not implemented)
5. `MergedOccurrence`

- 6. MergedAssociation
- 7. MergedAssociationRole
- 8. MergedScopeable (abstract class)
- 9. MergedScope
- 10. MergedReifiable (abstract class)
- 11. MergedTopicMapConstruct (abstract class)

Each of these classes implements the appropriate read-only TMDM interface.

In a Model-View-Controller design, this layer contains an internal view on (a set of) other topic maps, each allowed to ignore the merging rules. Each time a viewed topic map (e.g. a `BasicTopicMap`) changes in some aspect, an event is fired and the merged topic map is updated accordingly.⁸

During the update, the merged topic map itself may fire events to its downstream event listener. For example, it may fire an event stating that two formerly separate `MergedTopicMapConstructs` have now been merged. An application may use these notifications to update its user interface accordingly.

The Merged layer is only a view. Consequently, it does not need to modify its upstream `TopicMapConstructs`. Thus, it only needs to operate on a read-only version of a topic map, and consequently it requires the objects it is operating on only to implement the read-only TMDM interfaces layer, not necessarily the read-write TMDM interfaces layer. As the Merged layer is a view, it also only implements the read-only TMDM interfaces layer itself.

Representation. Each `MergedTopicMapConstruct` is internally represented as a list of the individual upstream `ReadableTopicMapConstructs` (this list is called components), together with the reference to the `MergedTopicMapView` (see below) and the key (see below) of the `MergedTopicMapConstruct`.

Merging topics. Most of the supplementing indexing information for a particular `MergedTopicMap` is stored in a `MergedTopicMapView` object, which is attached to every `MergedTopicMapConstruct` of that `MergedTopicMap`. One of the indexes is `itemIdentifierOrSubjectIdentifierToMergedTopicMapConstruct`, containing a mapping from `Locators` to `MergedTopicMapConstructs`. Each time an upstream `ReadableTopicMapConstruct` receives an additional item identifier and, similarly, each time an upstream `ReadableTopic` receives an additional subject identifier, the corresponding `MergedTopicMapConstruct` is registered in this index under the additional identifier. If, for this additional identifier, there already exists an entry, then merging is triggered. Equality of subject locators is

⁸ For example, consider that a new upstream `ReadableTopic` is created. Then, an event is fired to the downstream `MergedTopicMap`. Then, a new `MergedTopic` is created.

handled in the same way. Currently, merging of topics due to equality in the “reified” property is not implemented.

Merging statements. For each statement, there is a key object which represents that statement's equivalence class as defined by the TMDM. If two key objects are equal in each field, then these key objects themselves are equal. The choice of the fields of the key classes is guided by the TMDM's equality rules. For example, the data structure for the key for a `MergedOccurrence` is defined as follows:

```
public class MergedOccurrenceKey extends MergedScopeableKey {
    protected MergedTopic parent;
    protected MergedTopic type;
    protected Locator    datatype;
    protected String     value;
}
```

Whenever a statement is created or modified, an appropriate key object is entered in an appropriate index within the `MergedTopicMapView` object. If there is already an existing key object in the index which equals the new key object, then the statements of both keys are equal, and merging is triggered.

Dependent merging. If a `MergedTopic` is merged, then all the objects which are referencing this topic have to be updated. Thus, each `MergedTopic` maintains inverted indices about themselves, that is, sets of `MergedTopicMapConstructs` which, for some property, point to that `MergedTopic`; each set for one particular property. In case of merging, these sets are traversed and the values for that property for the dependent `MergedTopicMapConstructs` are updated accordingly. (This also means that their keys are changed to reflect the new value for that property, which in turn can lead to more merging.)

In the current implementation, these sets are not complete: They are only implemented for the properties `Association.type`, `AssociationRole.type`, `AssociationRole.player`, `TopicName.parent`, `Occurrence.parent`. Thus, such sets are missing for example for `TopicName.type`, `Occurrence.type` as well as for scope. Note that it is reasonable to assume that most of these sets are empty for most topics, as most topics are never used as an association type, association role type, occurrence type or topic name type. Thus, it should be more memory efficient to replace these sets, currently 4 (and later 7) per `MergedTopic`, either by appropriate indices in the `MergedTopicMapView` object or by a unified full inverted index (that is, exactly one set per `MergedTopic`, where each entry is a pair of a particular `MergedTopicMapConstruct` and the property within that particular `MergedTopicMapConstruct` which points to that `MergedTopic`). Implementing and evaluating this is left for future work.

Merging complexity. Merging two `MergedTopics` into one is quite similar to the union-find class of algorithms (employed for example in some implementations of Kruskal's algorithm): in both cases, two connected components are to be merged into one. The choice of what to merge with what may have a remarkable effect on the performance. Consider a list of n `MergedTopics`, each initially representing only 1 `BasicTopic`. Consider that, for some reason (e.g. adding subject indicators), all topics are being merged with each other, one after another, such that each time, the last two topics are merged. What if, at each step, both `MergedTopic` objects are deleted and a new `MergedTopic` representing the two is created instead⁹? Then both lists of individual upstream `BasicTopics` of both old `MergedTopics` have to be copied into a unified list of the new `MergedTopic`, yielding $O(n^2)$ copy operations. What if one `MergedTopic` object is reused and the other `MergedTopic` object is merged into it? If, at each step, the last topic is merged into the second but last topic, then still the number of copy operations is in $O(n^2)$. However, if at each step, the second but last topic is merged into the last topic, the number of copy operations is in $O(n)$. Thus, choosing the order of what to merge into what is important.

The weighted-union heuristic [Galler1964][Hopcroft1971][Fischer1972] teaches to always merge the smaller `MergedTopic` (the smaller connected component) into the larger one. Then, the number of copy operations is in $O(n \cdot \log(n))$, regardless of the initial number of `BasicTopics` in each `MergedTopic`. The proof is similarly straightforward: There are at most n initial `BasicTopics`, and each `BasicTopic` undergoes only about $\log_2(n)$ copy operations. Let $c(m)$ be the number of `BasicTopics` that a `MergedTopic` m contains. Suppose a `BasicTopic` b , directly before undergoing a copy operation, belongs to a `MergedTopic` m_0 , which is going to be merged with `MergedTopic` m_1 . This results in a new `MergedTopic` m_2 . Then, the equation $c(m_2) \geq 2 \cdot c(m_0)$ holds. The reason is that $c(m_2) = c(m_0) + c(m_1)$ and $c(m_1) \geq c(m_0)$. (If this was not the case, then the `BasicTopics` of m_0 would not be copied, but the `BasicTopics` of m_1 would be copied instead, which contradicts the assumption.) Thus, after each copy-operation of a `BasicTopic`, the size of the `MergedTopic`, which the `BasicTopic` is member of, has at least doubled. After k such steps, the `BasicTopic` belongs to a `MergedTopic` which has at least 2^k `BasicTopics`. Let $k_0 = \min(k \mid 2^k \geq n)$. After k_0 steps (possibly earlier), the `BasicTopic` belongs to a `MergedTopic` which has at least n `BasicTopics`. At this stage, no further merging is possible (because there is only one `MergedTopic` left, which contains each of the n `BasicTopics`). Thus, after about $\log_2(n)$ copy operations ($k_0 \leq \text{ceil}(\log_2(n))$) for each `BasicTopic`, the merging process is finished. ■

⁹ as suggested by the TMDM

The merging complexity considerations for other `MergedTopicMapConstructs` are similar.

Note that the conceptually simpler implementation may not always be the faster implementation. When merging `MergedTopic` m_0 with `MergedTopic` m_1 into a new `MergedTopic` m_2 , then, conceptually, m_0 and m_1 have to be removed from the indices and m_2 has to be inserted into the indices (see [ISO13250-2], section 6.2). However, now that we know that merging *into* an existing `MergedTopic` m_1 is faster, we also know that the address of m_2 equals to the address of m_1 (although the state at the address changes from m_1 to m_2). Thus, we do not need to remove (the address of) m_1 from the indices, because all pointers to the state m_1 later point to the state m_2 . We just have to remove everything pointing to the address of m_0 from the indices and insert new index entries such that they now point to the address of m_2 . If removing and inserting can be combined into one update operation, this is even better. The TMDM backend was initially implemented without that optimization, but is now available with this optimization – with tremendous speedups (see section 3). The reason for this speedup is the change of the complexity class: Consider the merging of n topic maps into a big topic map, one at a time, and consider a subject which is represented in every such topic map (e.g. by a topic which serves as a type topic for some common type of association or occurrence). Then, the corresponding `MergedTopic` has many `BasicTopics` as components. Each time a component is added to the `MergedTopic`, without the merging optimization, all components are considered (e.g. to list all subject identifiers) when unindexing and reindexing. Thus, the number of consideration operations is in $O(n^2)$. With the merging optimization, only the newest `BasicTopic` is considered at a time, thus the number of considerations is in $O(n)$.

Unmerging complexity. The design of the Merged layer, as a view to process whichever input it is confronted with, allows not only for merging, but also for unmerging. Consider that a property of a `BasicReifiable` is changed. In this case, the corresponding `MergedReifiableKey` changes, the `BasicReifiable` may be removed from the set of components of one `MergedReifiable` and it may be added to the set of components of another `MergedReifiable`.

However, this simple response to change may be more complicated for topics. Consider a bipartite graph, where `BasicTopics` and locators are vertices and where there is an edge between a locator and a `BasicTopic` if, and only if, the locator is a subject identifier of the `BasicTopic`. Then, for each connected component in the graph, all the `BasicTopics` which are member of that component should be merged. Now, consider that an application removes such an edge of the connected component (for example by executing something like

`basicTopic.removeSubjectIdentifier(someSubjectIdentifier)`). Then, the `MergedTopic` of that connected component should split iff the connected component splits. However, there is no straightforward way to determine (locally) whether a removal of just one edge makes a connected component split. For example, if the connected component is a large cycle, removing one edge does not make the connected component split. However, if the connected component is nearly a large cycle with just one remote segment missing, removing one edge makes the connected component split.

Thus, until a more thorough analysis of this problem is performed, the current implementation for splitting is to reduce splitting to merging by *atomicizing* the connected component. That is, all edges are removed and then all edges, except the one to be initially removed, are re-added, eventually resulting in either one connected component, or two. As this is an $O(m+n \cdot \log(n))$ operation (where m is the number of edges and n is the number of vertices of the connected component), this operation is quite expensive. It remains a question of future research whether this operation needs more optimization (like an aggregated “remove all locators at once” operation), as removing locators may not be a very common operation.

2.5 TM4J1 compatibility layer

This layer contains interfaces for wrapping TMDM objects into TM4J1 objects, all within the package `org.tm4j.topicmap.tm4j1`:

1. `TopicMapImpl`
2. `TopicImpl`
3. `BaseNameImpl`
4. `VariantImpl` (currently not implemented)
5. `OccurrenceImpl`
6. `AssociationImpl`
7. `MemberImpl`
8. `ScopedObjectImpl` (abstract class)
9. `TopicMapObjectImpl` (abstract class)

At the core of the wrapping layer, there is a `BasicTopicMap` together with a `MergedTopicMap` within each `TopicMapImpl`. All read accesses which should also take into account merged topics (which is the default) are forwarded to objects of the Merged layer, while all write accesses are forwarded to the objects to the Basic layer.

All necessarily translations between the different models (TMDM vs. TM4J1 data model) are done on the fly, on a best effort basis. In particular, multiple players per role (allowed in TM4J1) are not supported, nesting of variants (allowed in TM4J1 due to a misunderstanding of XTM 1.0) will not be supported (variants are not yet implemented), topic name types and occurrence data types (allowed in the TMDM) are inaccessible via the TM4J1 layer.

Instances of the compatibility layer (e.g. instances of `TopicImpl`) are created on demand. This may result in two different `TopicImpl` objects representing the same `BasicTopic`. But because they also represent the same `MergedTopic` and topic maps explicitly are designed for merging, this did not have an apparent negative effect so far.

2.6 TopicMapEventListener

The event handling model has been changed radically. In TM4J1, a JavaBeans `PropertyChangeListener` or a JavaBeans `VetoableChangeListener` was registered against a particular property of interest of a particular object of interest. The property of interest was indicated by supplying the name of the property as a string.

This has multiple disadvantages. First, using strings, instead of compiler-checked literals (e.g. Java enums), is error prone, as accidental misspellings do not result in compile errors, but are silently accepted. Second, using strings leads to string comparison at runtime, which is slow compared to mere pointer dereferencing which would be employed by the runtime environment if compiler-supported language constructs would be used. Third, the list of event listeners (or even just the pointer to this list) in every single `TopicMapObject` contributes to the memory footprint of these `TopicMapObjects`. Fourth, an event handling model based on `PropertyChangeListener` requires the event source to provide an old and a new value of the property to the listener. In case the property is an array and the event is to add an element to the array, this means that an array representing the old list of elements and an array representing the new list of elements have to be provided to the listener, at the same time. This does not only mean that the listener has to do a side-by-side comparison of both supplied arrays (which is at least an $O(n)$ operation) in order to find out which element has been added. This also means that just preparing for sending an event is not an $O(1)$ operation anymore, but an $O(n)$ operation, involving creating a copy of an array.

For example, the relevant code to add an association role to an association in TM4J1 looks like (“association role” is called “member” in the TM4J1 data model):

```

Collection oldMembers = (m_members == null)? Collections.EMPTY_LIST
                        : m_members;

// Fire vetoable change notification
Collection newMembers = new ArrayList(oldMembers);

newMembers.add(member);
fireVetoableChange("members",
    Collections.unmodifiableCollection(oldMembers),
    Collections.unmodifiableCollection(newMembers));

m_members = newMembers;
((MemberImpl) member).setParent(this);

firePropertyChange("members", oldMembers, m_members);

```

whereas the relevant code to add an association role to an association in TM4J2 looks like:

```

roles.add(role);
getEventListener().notifyAssociationRoleCreated(
    getContainingTopicMap(), this, role);

```

It is clear that the latter code is not only shorter, but reasonably expected to be faster, too.

Thus, the event handling model of TM4J2 has been redesigned radically compared to TM4J1.

First, the new event handling model does not use string constants. It also does not use enum constants. Instead, it models every event as an action, that is, in the Java language, a method call. Using method calls instead of event objects has the advantages that there is no need to create, write to, read from, or delete event objects. Instead, all of these actions happen implicitly on the stack. Another advantage is that there is no need to define different event classes for different types of events, which would have led to an event class zoo. A further advantage of the implicitness of method calls is the good support for optimization by current Java virtual machines: If it can be determined (by the JavaVM) that the only possible implementation of a method call is the empty implementation, then the method call itself, along with all preparations to calculate the arguments of the method, can be optimized away.

Second, there is only one, and exactly one event listener per topic map, and there is no event listener for any of the other `TopicMapConstructs`. First, this saves main memory in all but one objects of each topic map. Second, by settling for exactly one event listener (and not for zero or one, or, respectively, a variable number of event listeners), an explicit check for null pointers, or, respectively, a

for-loop, is avoided, making the code more readable and less complex and thus more amenable to automatic optimization. Third, the default event listener is the no-operation event listener. If the JavaVM detects the no-operation-nature of the event listener (and chances are that current JavaVMs do so), then, as mentioned above, event handling is a no-operation also on the caller side, and thus with zero cost with respect to CPU cycles.

This does not mean that multiple event listeners are not supported. The concern for multiple listeners has just been separated from the concern of calling “the event listeners”: If multiple event listeners are desired, it is easy to create a multiplexer event listener which forwards each event it receives itself to multiple event listeners.

2.7 Architectural overview

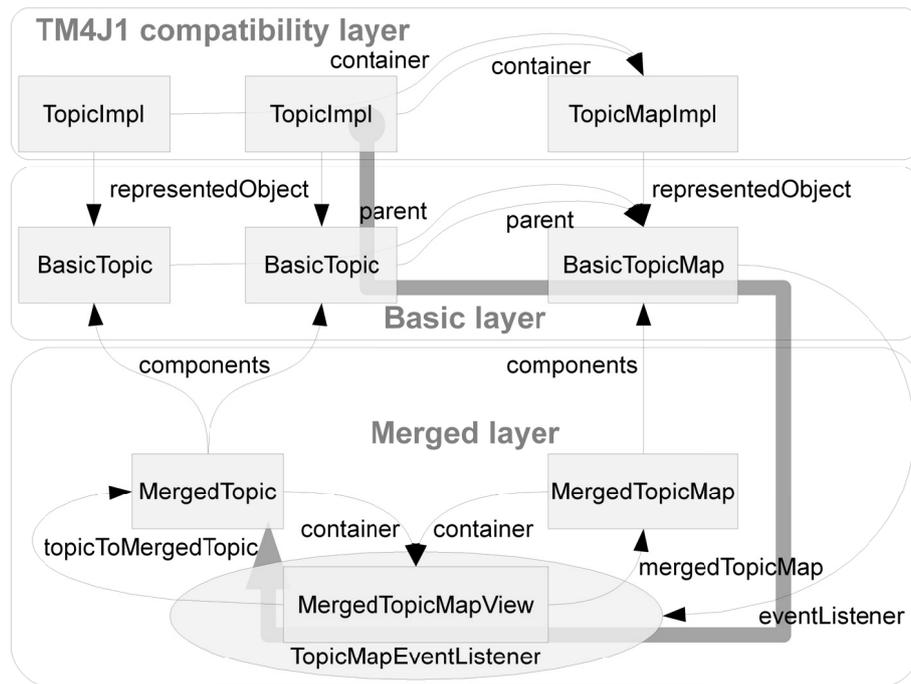


Fig. 1. Architectural overview

Putting everything together, Figure 1 shows a simplified example object graph and the three layer. The path throughout the graph shows an example event handling path.

2.8 XTM 2.0 reading support

XTM 2.0 reading support has been implemented partially, currently as what is called colloquially as a “hack”. That is, element names of XTM 2.0 are handled in the same way as the corresponding element names of XTM 1.0 are handled. Features of XTM 2.0 which are not available in XTM 1.0 are thus silently ignored, even if the final backend supports the TMDM, because up to now, there are no “feature” holes punched into the curtain of the TM4J1 API, even if both sides of the curtain support more modern topic maps processing. A proper XTM 2.0 reading and writing support, building directly on TMDM backends, is part of the future work.

3 Evaluation

The contribution to the TM4J project has a size of more than 9000 lines of code, and it is available in the current CVS tree of TM4J, open for public review. We have benchmarked the current “TMDM” backend prototype with merging optimization and the current “TMDM” backend prototype without merging optimization against the old “memory” backend on a testing machine, containing 8 Intel Xeon E5335 cores and 16GiB of RAM, running Linux 2.6.25 in 64-bit mode. We take a set of between 1 and 1024 small XTM 2.0 files (on average 111 topic map constructs (among them about 24.6 topics and 22.5 binary associations) per file) generated by yet-to-be-published software out of the DBLP¹⁰ dataset, and let all these backends merge these files into one merged XTM. Some topics are present in all files, most topics are only present in one or two files. Most associations are present in one or two files. Most topics have 2 or 3 subject indicators. We measure the processing time as well as the maximum used memory (by using statistical output of the garbage collector). We use the “Java HotSpot(TM) 64-Bit Server VM (build 10.0-b22, mixed mode)”, the Java command line options “-Xmx8G -da” for processing time tests and the additional command line options “-verbose:gc -XX:MinHeapFreeRatio=2 -XX:MaxHeapFreeRatio=4 -XX:MaxNewSize=2048k ” for RAM tests (to enable frequent garbage collector statistical output). All tests have been performed with pre-warmed caches to minimize influences of e.g. disk latency.

¹⁰ DBLP is one of Computer Science's large bibliographic databases. One XTM 2.0 file corresponds to the author-paper relationship shown for one particular author, for example http://dblp.uni-trier.de/db/indices/a-tree/k/Knuth:Donald_E=.html. (Thanks to MICHAEL LEY for providing access to the software which generates all DBLP author's pages.)

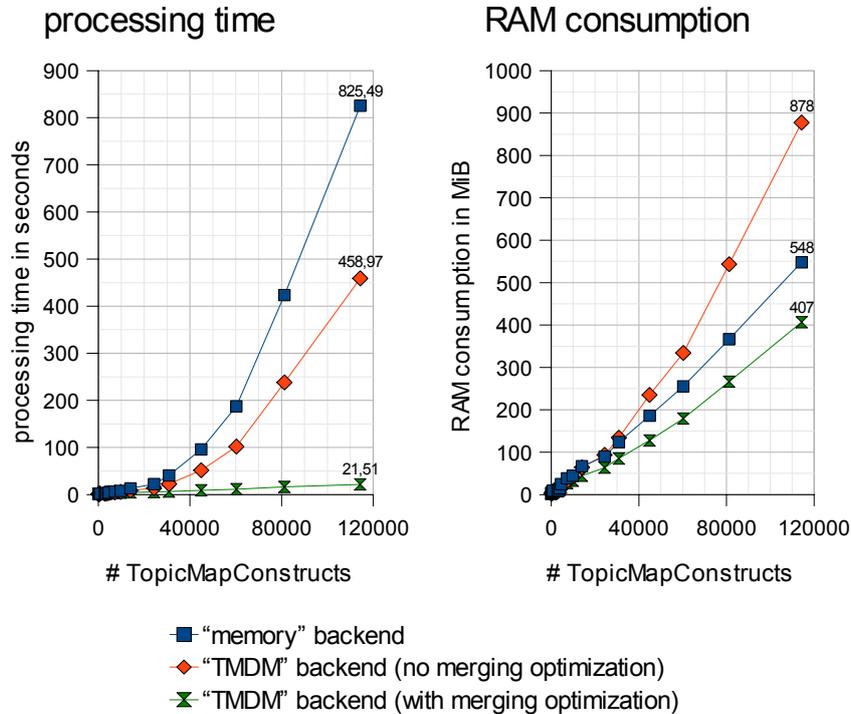


Fig. 2. Synthetic benchmark of merging over different input sizes

While the unoptimized “TMDM” backend prototype already outperforms the “memory” backend, it is evident that the merging optimization has a crucial impact, providing an about 20-fold increase in merging performance. The parabola shape of the processing time graph of the “TMDM” backend (without merging optimization) is well in line with the theoretic considerations that, without the merging optimizations, the merging complexity for n topics to be merged into one is in $O(n^2)$. The shape of the respective graph of the “memory” backend suggests that this backend, too, could profit from the merge optimization.

The “TMDM” backend prototype with merging optimization consumes less memory than the “memory” backend. However, this may also partly be because the “TMDM” backend prototype does not yet implement all desired features (e.g. support for variants, and more indices).

Interestingly, the memory footprints of both the “TMDM” backend prototype variants differ considerably, which should not happen for a memory-neutral

optimization. This could be an indicator of a memory leak in the “TMDM” backend prototype, and needs further investigation.

In general, the memory usage for an implementation employing standard Java techniques is still disappointing: about 3730 bytes per `TopicMapConstruct`. For a comparison, the corresponding XTM 2.0 input files consume just about 163 bytes per `TopicMapConstruct`. A quick analysis using the “jmap” tool¹¹ revealed that not only a big part of the memory consumption is due to storing characters in UTF-16 format (each character consuming 16 bits), but also that an even bigger part of memory consumption is due to instances of `java.util.HashMap$Entry` and arrays thereof. Thus, not only changing the internal string encoding to more space efficient encodings (like UTF-8) or employing string compression techniques (as most locators happen to have common prefixes), but also changing the implementation of `java.util.Map` from `java.util.HashMap` to more space-efficient ones as well as replacing small maps (e.g. those with only one or 2 entries) by specialized, compact data structures, seem to be promising improvements.

4 Future Work

Bock raised in [Bock2008] the issue of using a Domain Specific Language to represent the TMDM, and of using interpreters of this language to generate each of the sets of classes or interfaces of the TMDM backend of TM4J2. This is a very interesting approach, as it not only helps to avoid coding errors, but also helps to change deep architectural decisions on a whim. For example, we expect that instead of implementing locators as `Locator` objects, but as UTF-8-encoded `byte[]` strings, the resulting topic map engine would both have a smaller memory footprint and be faster. (At the same time, the source code would lose object oriented elegance, which, however, is acceptable if the source code is automatically generated.) By tuning different architectural decisions, one can create instance specific Topic Maps engines, i.e. Topic Maps engines which are suited for a very specific type of topic map¹², to the point where finding the fastest Topic Maps engine for a particular topic map is a classical optimization problem. To make this possible in the first place, as many parts of TM4J2 as possible have to be reformulated in terms of a DSL. (Bock has already completed the formulation of the TMDM and the automatic generation of the read-only TMDM interfaces, the read-write TMDM interfaces, the Basic layer and the event listener interface.)

¹¹ See <http://java.sun.com/javase/6/docs/technotes/tools/share/jmap.html>

¹² For example, a topic map containing Chinese text data would suffer from UTF-8 encoding.

Some statistical assumptions about “typical topic maps” need to be verified empirically. Among them are the following:

1. The set of distinct scopes is much smaller than the set of scopeables.
2. The set of topics used as association type is small.
3. The set of topics used as association role type is small.
4. The set of topics used as topic name type is small.
5. The set of topics used as occurrence type is small.

MAICHER suggested (in private conversation at the TMRA2007) that late, on demand merging in federated Topic Map databases may be a good thing, because it allows individual member Topic Maps to be virtual. This idea has merit, as those Topic Maps implementations also do not need to support an update notification (event generation mechanism). It also avoids the unmerging performance problem, because on demand merging does not require unmerging at all. Furthermore, supporting late merging is actually a quite natural way for supporting federated Topic Map databases in the first place. Thus, a separate `org.tm4j.topicmap.tmdm.merged.ondemand` view is part of future work.

Apart from dynamic early merging and dynamic on-demand (late) merging, support for static merging (that is, directly between `BasicTopicMapConstructs`) may be implemented. However, if the dynamic early merging layer is efficient enough, there may not be much incentive to implement static merging functionality.

The backend shown here is for RAM-only storage.¹³ Exploring the extension of the backend to disk-based storage (e.g. using JDO) has been started by the authors, but is not part of this paper.

Acknowledgements

Thanks go to the Topic Maps Lab¹⁴ at the University of Leipzig, Germany, for kindly providing access to the testing machine.

¹³ Due to the gap between RAM latency and disk latency widening (and RAM having a lower price per (random) access than disk) and due to the price of RAM declining, we believe that distributed RAM-only Topic Maps engines will play an important role in the future.

¹⁴ See <http://www.topicmapslab.de/>

References

- [ISO13250-2]: International Organization for Standardization/International Electrotechnical Commission — Joint Technical Committee 1 — Subcommittee 34 — Working Group 3: “ISO/IEC IS 13250-2:2006: Information Technology — Document Description and Processing Languages — Topic Maps — Data Model” International Organization for Standardization, Geneva, Switzerland (August 2006)
<http://www.isotopicmaps.org/sam/sam-model/>
- [ISO13250-3]: International Organization for Standardization/International Electrotechnical Commission — Joint Technical Committee 1 — Subcommittee 34 — Working Group 3: “ISO/IEC IS 13250-3:2007: Information Technology — Document Description and Processing Languages — Topic Maps — XML Syntax” International Organization for Standardization, Geneva, Switzerland (August 2006)
<http://www.isotopicmaps.org/sam/sam-model/>
- [XTM1.0]: TopicMaps.Org Authoring Group: “XML Topic Maps (XTM) 1.0” International Organization for Standardization, Geneva, Switzerland (August 2001)
<http://www.topicmaps.org/xtm/1.0/>
- [TMAPI1.0SP1]: Kal Ahmed, Lars Marius Garshol, Geir Ove Grønmo, Stefan Lischke, Lars Heuer, Graham Moore: “TMAPI — Common Topic Map Application Programming Interface — 1.0 SP1” (February 2005)
<http://www.tmap.org/>
- [Galler1964]: Bernard A. Galler, Michael J. Fisher: “An improve equivalence algorithm” in *Communications of the ACM*, volume 7, issue #5, pages 301..304 (May 1964)
<http://portal.acm.org/citation.cfm?doid=364099.364331>
- [Hopcroft1971]: John E. Hopcroft, Jeffrey D. Ullman: “A linear list merging algorithm”, Technical Report number CS-CSD-71-111, Cornell University, Ithaca, New York, USA (November 1971)
<http://ecommons.library.cornell.edu/bitstream/1813/10810/2/TR71-111.pdf>
- [Fischer1972]: Michael J. Fischer: “Efficiency of equivalence Algorithms”, Artificial Intelligence Memo number 256, A. I. Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA (April 1972)
<http://dspace.mit.edu/bitstream/1721.1/6201/2/AIM-256.pdf>
- [Bock2008]: Benjamin Bock: “Topic Maps Middleware”. Master's thesis, University of Leipzig, Germany (May 2008)
http://academia.bock.be/publications/Bock2008_Topic-Maps-Middleware.pdf

ActiveTM: A Topic Maps – Object Mapper

Benjamin Bock

University of Leipzig, Johannisgasse 26, 04103 Leipzig, Germany

bb--tmra2008-activetm@bock.be

Abstract. Currently, the most common way to programmatically access Topic Maps data is the use of a Topic Maps API, like TMAPI. Another approach, besides the use of a query language like TMQL, is the encapsulation of the Topic Maps related code in domain-specific model classes. This concept is similar to object-relational mapping (ORM) which encapsulates access to a relational database inside the model classes. These techniques decouple the data store specific code from the business logic. For ORM, there are several prevalent design patterns, most notable the Active Record pattern by Fowler. For Topic Maps, no such pattern is established. This paper introduces Active Topic Maps, a pattern for Topic Maps – object mapping, the domain-specific language ActiveTMML to define such a mapping, and a prototypical implementation, called ActiveTM. ActiveTM is based on Ruby Topic Maps and also supports the generation of web-forms based on ActiveTMML definitions. This full-featured software stack greatly improves the development productivity of Topic Maps based portals compared to other solutions.

1 Introduction

The Topic Maps Data Model (TMDM) [1] offers many liberties while designing an ontology. Many classes and methods are required to offer the full flexibility and functionality of the TMDM to a programmer using a Topic Maps engine with a generic application programming interface (API), e.g. Ruby Topic Maps (RTM) [2]. The parameters of many of these methods are manifold. This is because the Topic Maps constructs represented as instances of the classes have many properties to be retrieved and modified. All these methods are aligned to the TMDM and not optimized for the particular domain. The development of a domain application requires the programmer to either use the generic TMAPI-

like methods directly or to encapsulate them in domain model classes. Here, TMAPI does not only refer to the Java- and PHP-based standardized APIs [3] [4] but to any Topic Maps API with a similar set of functions.

The encapsulation in domain model classes allows to use only the model objects in the other parts of the application because the program code to access the data store once resides solely in the model. This technique is commonly referred to as Model-View-Controller (MVC) pattern [13]. The creation of these model classes is straightforward from the definition of an ontology but still requires some amount of work. The idea presented here is to define the ontology in a domain-specific language (DSL) [6] and use this to generate the model classes, including the code to retrieve and persist the objects in a Topic Maps data store.

The prevalent functions of persistent storage are create, read, update, and delete (CRUD) [14]. Create and update include scrutinizing the constraints for the particular data objects. Using TMAPI directly does not allow to check particular constraints of the domain model. The consistency of a topic map can be verified afterwards using a custom constraint language or eventually with Topic Maps Constraint Language TMCL [12]. Deletion of data objects may be restricted due to other constraints or may entail other deletions or updates. In relational databases, these functions may trigger functions to preserve consistency. For Topic Maps, no standardized approach exists. Additionally, access control based on data in the topic map is needed in real world applications but not provided in a standardized way by current solutions. Access control is not covered in this paper.

2 Previous Approaches

The most commonly used technique to access Topic Maps data is the usage of a library with a TMAPI-like interface. A goal is to encapsulate and thus simplify the usage of a TMDM data store using a special API. The next subsections illustrate the usage of TMAPI and two previous approaches to optimize the way how Topic Maps data is accessed and how a domain can be modeled.

The technique object-relational mapping (ORM) is, besides using SQL, the predominant way to access relational databases. The popular Active Record design pattern implements ORM. The homonymous Ruby library offers a DSL to describe domain model classes and transparently implements the database access for these classes.

Bringing together these two techniques finally leads to the concept of Topic Maps – object mapping.

2.1 A Basic Read Operation using TMAPI

The following example illustrates the steps necessary using Java TMAPI 1.0 to read a certain name from a topic `t`. It does not even include handling of scopes but it is already quite lengthy.

```
// get typing topic (pseudo code)
Topic type = tm.getTopicBySubjectIdentifier
    ("http://psi.example.com/firstname");

// iterate over all topic names

Iterator i=t.getTopicNames().iterator();
while (i.hasNext()) {
    TopicName tn = (TopicName) i.next();
    // check type
    if (tn.type == type) {
        // use name, e.g. output it
        System.out.print( tn.getValue() );
        break;
    }
}
```

HEUER introduces the concept of accessing characteristics of a topic using a Hash-like syntax in the Topic Maps engine *Mappa* [5]. Transferring this concept to the Java language, the previous example would look like this¹:

```
Set<TopicName> names = t.get
    ("-http://psi.example.com/firstname");
for (TopicName tn : names ) {
    System.out.print( tn.getValue() );
    // break after the first one
    break;
}
```

This is significantly shorter than the first example. Using it in Python, the language *Mappa* is written in, is even shorter as Python's Syntax is more terse than Java's. In Ruby Topic Maps, the Hash-like access works the same way as in *Mappa*:

```
puts t["-http://psi.example.com/firstname"].first.value
```

Still, the way to access the data is not domain specific. The subject identifier in the string cannot be checked by a compiler nor by an interpreter at runtime. Assuming the topic `t` represents an object `p` of class `Person`. There should be two methods in `p`: one to get and one to set a single first name. Depending on the

¹ This implicitly assumes an API using Java generics

domain ontology (where multiple first names may be allowed), this could also be methods to get and set a list of first names and additionally to add and remove single first names from this list.

2.2 Topic Maps Objects

MOORE, AHMED, and BRODIE demonstrated *Topic Map Objects* (TMO) at the TMRA 2006 conference [16]. TMO is a framework providing domain-specific classes to retrieve and update Topic Maps data in a distributed environment. MOORE, AHMED, and BRODIE don't build upon TMAPI but on *Topic Map Webservices* (TMWS). TMWS provides access to a topic map using a SOAP interface. The goal of TMO is to unify the advantages of TMWS with the features of modern object-oriented languages like Java and C#. The resulting framework allows programmers to work with domain objects without knowing the TMDM in detail.

TMO consists of two components: The first component of TMO is TMWS. The TMWS framework used is functionally equivalent to TMAPI. In this component, no higher level of abstraction or domain-specificity is introduced. From the perspective of abstraction, the feature of transactional updates is not relevant, however this could be exploited to integrate constraint checking. The intention of this feature seems to be optimization of network traffic. The Object Manager Service (OMS) is the second component of TMO. This component can create domain-specific objects from Topic Maps data and provide an application with these objects in a serialized fashion. The topic maps ontology data is part of the class definition while the instance data resides in the object. The object manager contains the program code to read all properties of the domain object from the topic map and fill the private variables in the objects at the time of its construction. The updates in the objects can be transferred back to TMWS later. The domain classes contain the program code to update the objects, later read accesses see the current values.

TMO is written in the C# programming language for the .NET platform. It is part of the commercial product TMCORE by Networked Planet Limited, Oxford, UK, to which the authors belong [at the time of writing]. A graphical user interface, based on Microsoft Visual Studio is planned² but not publicly available yet. It involves creation of an XML document which makes the annotation of domain classes unnecessary. The automatic creation of program classes seems not to be planned, so one has to assume that the code to update a topic map has to be written by hand. The following example shows the definition of a class `Person` with a property `firstname`.

² According to MOORE, in a private conversation on 2008-04-06

```

[TopicTypeAttribute("http://www.networkedplanet.com/person")]
public class Person : TopicMapObjectBase {
    private string m_name;
    private string m_age;

    [TopicNameAttribute()]
    public string FirstName {
        get { return m_name; }
        set {
            OccurrenceSet(this, "FirstName", value);
            m_name = value ;
        }
    }
    [TopicOccurrenceAttribute
    ("http://www.networkedplanet.com/ otypes/age")]
    public string Age {
        get { return m_age; }
        set {
            OccurrenceSet(this , "Age" , value);
            m_age = value ;
        }
    }
}

```

The example is derived from one of the examples given in [16] and allows to reason that TMO uses a pre-TMDM data model³, topic names do not have a type yet. The occurrence age shows how the type would be specified. Another observation is that there is no clear distinction between `OccurrenceSet` and `TopicNameSet`. This might be a typo in the document, though.

A graphical user interface would clearly be an advantage of this solution, while the usage of a web service may lead to performance deficits compared to the usage of a local library's API. TMO objects can only be used asynchronously, a direct update of the underlying topic map is not possible with this architecture. The domain-specific information is given at (at least) two locations: the object manager (for reading) and the domain classes (for updating).

2.3 Bogachev's Subject-Centric Programming Language

In [8], BOGACHEV presents the similarities of Topic Maps and the COBOL programming language. The advantage of COBOL is the definition and manipulation of *business data* in the language. In many modern programming languages this domain specific information was outsourced to a relational

³ At the time of writing of TMO, TMDM was not finalized, so in fact this is not a big surprise.

database, decreasing transparency and simplicity. With these assumptions in mind, BOGACHEV developed his subject-centric programming language [9].

He criticizes that object-oriented languages help to model things on a computer, but not to represent knowledge about these things. He questions what happens if information changes over time and how to deal with information from different sources. Furthermore, he asks how interference rules and calculated values can be respected in such a system and how visibility and update rights can be bound to specific user groups.

To address all this, he defines *metaproperties* which are classes derived from a specific property type. In the example, the property `firstname` is derived from `ActiveTopic::Name`⁴.

```
class FirstName < ActiveTopic::Name
  psi      'http://psi.ontopedia.net/firstname'
  historical true
  card_max 1
  domain   :person
end

class Person < ActiveTopic::Topic
  psi      'http://psi.ontopedia.net/Person'
  name     :firstname
end
```

For both, the definition and the usage, BOGACHEV orientates himself at the syntax of the Active Record Ruby library, but there is no implementation⁵.

A continuative work is *Authoring topic maps using Ruby-based DSL: CTM, the way I like it* [10], a domain-specific language for defining Topic Maps ontologies and facts in a Ruby-based syntax. The emphasis here is not a programming framework but an alternative approach to the Compact Topic Maps Notation (CTM) [11].

2.4 The Active Record Design Pattern

FOWLER develops the design pattern *Active Record* [15] which implements the principle of object-relational mapping. The program code to access the storage layer (i.e. the relational database) is directly part of the model classes in Active Record. The objects are created or retrieved from the database using class

⁴ The choice of the namespace prefix `Active` for the class and the usage of a Ruby-based syntax make it obvious that he has the Ruby on Rails component Active Record in mind.

⁵ Private conversation, 2008-04-02

methods from the same class. They are stored using instance methods of the concrete objects.

The Ruby library *Active Record*⁶ is part of the web application framework *Ruby on Rails*⁷. It implements the homonymous design pattern. In Active Record, the names of the getters and setters for simple properties are derived from the column names in the database schema. They cannot be defined in the model classes and cannot be retrieved from the model classes without an active database connection. Thus, the complete model definition is available at runtime only. Associations between objects are defined using a DSL and not automatically derived from the database schema using e.g. the foreign keys. For a model class without associations, a class extending `ActiveRecord::Base` is sufficient. The statements to define an association to another class are called `has_one`, `has_many`, and `has_and_belongs_to_many`. The opposite table (or the join table respectively) holding the foreign key needs to use the statement `belongs_to`. For all statements, there are parameters to refine the definition if the schema does not match the naming convention exactly. The example shows the definition of a class `Person` with some associations and the creation of a single instance. The exact usage can be found in the Active Record API documentation⁸.

```
class Person < ActiveRecord::Base
  belongs_to :home_country, :class_name => "Country"
  has_many :cars, :foreign_key => "owner_id"
end
p = Person.create
p.firstname = "Benjamin"
p.save
```

When calling the method `save` the library executes the following statement⁹.

```
INSERT INTO 'people' (first_name) VALUES ('Benjamin');
```

The Active Record library provides a second, separate DSL called *Migrations* to describe the database schema. Changes to the ontology always require changes to the database schema and must be reflected there. Thus, a restart of an application is needed whenever the ontology and consequently the schema changes.

⁶ <http://wiki.rubyonrails.org/rails/pages/ActiveRecord>

⁷ <http://www.rubyonrails.org>

⁸ <http://api.rubyonrails.org>

⁹ Please note the table name is “people” but the class name is “Person”. This is a convention used in the Ruby library Active Record, which contains a pluralization module. In the class definition this convention can be overwritten

3 Domain Modeling

The definition of an ontology is part of the modeling of the domain to which the application should be specific. Nowadays, software developers are used to model their problem using object oriented techniques. There are many tools available to aid such a development process, ranging from a sheet of paper and a pen to sophisticated UML [17] editors. Integrated development environments like Netbeans¹⁰ or Eclipse¹¹ directly assist writing code in a particular programming language. The result of a development process is a formal specification of the model, covering all relevant aspects to address the problem of the given domain.

Defining model classes using UML results in a class diagram from which domain specific code can be generated. The resulting code is self-contained and does not include a mapping to a Topic Maps ontology. Our goal is to model a Topic Maps ontology and the corresponding model classes at the same time. Generally, to allow an efficient workflow, it is essential to do exactly the things necessary and avoid everything else. Applied to modeling the ontology of a domain problem, this includes the description of the relevant entities, their characteristics and associations.

Using Topic Maps technology, the ontology is part of the topic map itself. The upcoming Topic Maps Constraint Language (TMCL) [12] strives to standardize the definition of ontologies in Topic Maps. However, this does not include naming of classes nor methods. For an ontology to model both, object-oriented and Topic-Maps-oriented aspects, TMCL has to be augmented or a new language has to be created. TMCL is not finalized at the time of writing and, following a pragmatic approach, the creation of a new language was chosen with ActiveTMML. As a later step, a formal mapping between ActiveTMML and TMCL should be defined. This could be done using a small ontology which defines the basic information necessary to create ActiveTMML code out of a Topic Maps ontology defined in TMCL.

Alternatively, TMCL fragments could be used as parameters to ActiveTMML statements. The benefit would be a single source for a complete ontology definition. The downside would be that this would presumably not integrate well with graphical TMCL editors.

¹⁰ <http://www.netbeans.org>

¹¹ <http://www.eclipse.org>

4 ActiveTM

ActiveTM is a Ruby library implementing ActiveTMML, the Active Topic Maps Modeling Language. In this section, firstly ActiveTMML will be introduced, then the library itself is presented. The library ActiveTM is not the only use case for ActiveTMML, as it can also be used as a basis for code generation in other languages.

4.1 ActiveTMML

ActiveTMML is an ontology modeling language for both, Topic Maps and object-oriented models in a single language. It does not (yet) strive to be feature-complete regarding the flexibility of the TMDM but to be functional for the *common 80% of use cases*. As ActiveTMML is only suitable for ontology modeling, it is called a domain-specific language (DSL) [6] [7]. DSL are commonly divided into two types: internal and external DSL. While external DSL come with their own syntax, internal DSL borrow their syntax from a host programming language. Consequently, internal DSL are constrained by their host language's syntax but they also benefit from their toolchain, i.e. can be compiled or interpreted using the host language's tools. This liberates the developer of an internal DSL from developing a parser and leaves him with adding semantics to the given syntax. ActiveTMML is an internal DSL using the host language Ruby¹². Compared to other popular programming languages like Java¹³, C#¹⁴, and Python¹⁵, Ruby offers a comparatively free syntax.

There are two flavors of ActiveTMML. The standalone syntax uses just method calls like `model` and `occurrence` in special contexts. The in-class syntax is used as part of a class definition, as it is done in Active Record, and will be detailed in the ActiveTM section. The following example shows the standalone syntax:

```
model :Person do
  name :firstname
  name :lastname
  occurrence :age
end
```

¹² <http://www.ruby-lang.org>

¹³ <http://java.sun.com/>

¹⁴ <http://www.ecma-international.org/publications/standards/Ecma-334.htm>

¹⁵ <http://www.python.org/>

This standalone ActiveTMML code technically calls a method `model` and passes two parameters: The symbol¹⁶ `Person` and a block of code, introduced by `do` and ended by `end`. This is common Ruby syntax and can be executed by any Ruby interpreter. Consequently, ActiveTMML code can be seamlessly mixed with other Ruby code.

The method `model` uses a special context in which the definition of this particular model is evaluated. A context is a class, module or object which implements methods corresponding to the statements of ActiveTMML. During the evaluation of the block (which is the definition of a domain model class), the calls are delegated to the context class, module or object.

The code of the method `model` and the context can be *anything*, depending of the concrete implementation of the ActiveTMML language. Possibilities range from generating classes (as done in section 4.2) to generating files (e.g. source code for a particular language or library, as proposed in section 5). It is also possible to produce any other output, for example a relational database schema and ActiveRecord classes based on the model. There are prototypes fulfilling exactly this purpose. Additionally, the output of a TMCL file is an option.

The obvious object-oriented interpretation of the example above is to create a class called `Person` with getters and setters for the properties `firstname`, `lastname` and `age`. The Topic Maps interpretation of the same definition is a topic identified by the item identifier¹⁷ “Person” typing other topics, its instances. The default identifier can be overwritten using the method `psi` in the code block. It is not possible to define PSIs of instances directly in ActiveTMML. An algorithm generating PSIs for instances depends on the concrete implementation of an ActiveTMML interpreter. Section 4.3 explains how this was solved in ActiveTM.

The example above defines three characteristics, two names and one occurrence. The argument to the methods `name` and `occurrence` is interpreted as an item identifier for the type of the characteristic. This can be overwritten using the keyword parameter `psi` in each method. The datatype for names is always string, for occurrences it defaults to string, too. The datatype of occurrences can be overwritten using the keyword parameter `datatype`.

A slightly bigger example shows the usage of keyword parameters in the Ruby syntax as well as the definition of a binary association. The definition of an association consists of a parameter for the name, the role type on this side of the

¹⁶ Ruby symbols are similar to LISP symbols. In short, a Ruby symbol is a word preceded by a colon. It is commonly used as a constant. Technically it is comparable to an internalized `String` in Java.

¹⁷ The item identifier is relative here. According to TMDM it must be resolved against a locator to make it absolute.

association, and the association type. The other role type is retrieved from the name of the association-property (in this case “country”), unless given as the fourth parameter. In natural languages, the type of the thing referred to, often¹⁸ as is the type of the opposite role in a Topic Maps ontology.

```
model :Person do
  name :firstname, :psi => "http://psi.example.com/first"
  name :lastname, :psi => "http://psi.example.com/last"
  occurrence :age, :datatype => "xsd:integer"
  has_one :country, "inhabitant", "country-inhabitant"
end
```

The statement `has_one` adds the constraint that there is only one country in the given association with this a particular person. The pendant to `has_one` is `has_many`. These two method names are inspired from the Active Record library, their parameters and interpretation differs due to the different intention of modeling.

An obvious feature to add to the ActiveTMML are model constraints, like defining cardinalities. This could be achieved using additional keyword parameters for example. The downside is the increasing complexity of both, the model code and the interpreter code. Once TMCL is finalized, it should be used to define finer granular constraints. For the implementation of ActiveTM, the same functionality is achieved using filters and validations as it is done in Active Record.

4.2 Definition of Model Classes in ActiveTM

Besides the standalone ActiveTMML code, a class definition in ActiveTM can be done using the standard Ruby syntax to define a class. Therefore, the class needs to be provided with the necessary code for the ActiveTMML statements. This can be done in two ways: by extending the superclass `ActiveTM::Base` (analogous to Active Record) or by including the module `ActiveTM::Topic` as a Mixin. The latter allows more liberties in terms of the class hierarchy.

```
class Person < ActiveTM::Base
  topic_map "http://psi.example.com/"
  psi "http://psi.example.com/ontology/person"

  name :firstname
```

¹⁸ An exception is for example the artificial language Lojban which was developed by the Logical Language Group in 1987. In this language, not the other role is addressed but the relation of the other thing to the current role. A child would refer to its “mother” (English term, natural referencing style) as “the one I am child of” (English term, Lojban referencing style), thus using the own role type.

```

names :middlesnames
name :lastname
occurrence :age, :datatype => "xsd:integer"
has_one :country, "inhabitant", "country-inhabitant",
        :class => :Country

def fullname
  "#{firstname} #{lastname}"
end
end

```

This creates a single class `Person`. The definition of the class itself is pure Ruby code. The statements `topic_map`, `psi`, `name` and so on are basically calls to methods in the class scope. The definitions of these methods are provided by the class `ActiveTM::Base` (or by the module `ActiveTM::Topic` respectively). Each instance of the class holds a single reference to a `Topic` in the underlying `Topic Map`.

The example above also introduces the method “`topic_map`”. This method defines the base locator for this class. The statement “`names`” introduces a characteristic which may occur multiple times. Another addition is the usage of the keyword parameter “`class`” with the symbol “`Country`” in the “`has_one`” statement. This specifies that the retrieved object should be interpreted as an instance of class `Country`, no matter what other types it is instance of.

The definition of the method `fullname` in the previous example shows the mixture of of normal Ruby code with the `ActiveTMML` code, allowing to create virtual properties based on the definition of existing ones. The number-sign and the curly brackets are Ruby string interpolation syntax. This allows to embed the results of the methods called directly in the string.

4.3 Usage of ActiveTM Objects

As with the definition, the usage of `ActiveTM` objects is aligned with `Active Record`. Until now, only the ontology layer was covered. For the usage, the instance layer comes into play. In the instance layer, referencing instance topics a requirement. Referencing topics works using identifiers (internal, subject identifiers or subject locators) for creating and querying particular topics. The language-internal object references serve all other purposes.

The following example shows the creation of a new `Person`-object. A parameter can be passed to the `create` method to define an identifier. If not given, an identifier will be generated. The default algorithm to generate an identifier is to append a random fragment identifier to the type `PSI`. This can be overwritten by

providing the class with a instance method `generate_psi` which acts as a hook. This approach is similar to the `before_save` hook in Active Record.

```
p = Person.create("johndoe")
p.firstname = "John"
p.add_middlename "George"
p.lastname = "Doe"
p.save
```

As shown in the example, for single characteristics, a setter and a getter are created, for multiple characteristics an add method as well as a remove method and a getter are created. The same principle applies to `has_one` and `has_many`.

Similar to Active Record, there is a default finder as well as dynamic finder methods for the characteristics. The default finder takes an identifier, the dynamic finders accept an argument like the setter methods. The example shows several ways to retrieve the single topic created above. There are also finders to find multiple objects instead of only returning only the first one found. As always, the usage follows the Active Record example.

```
p = Person.find("johndoe")
p = Person.find_by_firstname("John")
ps = Person.find(:all)
p = Person.find_all_by_firstname("John")
```

5 Code Generation

The methods to access the characteristics and association of topics follow a common scheme which can be formalized in a code template. In ActiveTM these templates are evaluated at runtime. ActiveTMML can also be used to generate code or other output in any language, given templates of code to fill the domain-specific parts in. The following example shows a simplified but working implementation of the `name`-method which creates a getter for the `firstname`-property. The comment below shows the code which is actually send to `eval`.

```
def name(property_name, options={})
  name_type = options[:type] || property_name
  eval <<-EOD
    def #{property_name}
      @topic[\"#\#{name_type}\"].first.value
    end
  EOD
end
#def firstname
```

```
# @topic["firstname"].first.value
#end
```

Additional to the generation of accessor methods, also meta information can be integrated into the definition of classes. Active Record creates a method called “columns” which introspects the schema and returns a list of database columns for this model object. The information about this columns can be used to generate so-called “scaffolds”, complete CRUD users interfaces for the specific model objects. They provide the developer with a basic user interface for free. This basic interface can be used for administrative purposes as well as the basis for the interface for end users.

6 Ontology Introspection

Besides the accessor methods for the characteristics and associations and besides the introspection methods, the classes also provide a getter `topic` which returns the underlying Ruby Topic Maps topic object. Using this topic, all aspects of the TMDM can be addressed. Setting the flag `acts_as_topic` enables the methods directly in ActiveTM objects, for example the set of occurrences:

```
# standard way
p.topic.occurrences
# direct way
class Person
  acts_as_topic
end
p.occurrences
```

The another flag, called `acts_smart` enables ActiveTM objects to look into the Topic Map and find possible characteristics for properties which are not explicitly defined. Assuming that for class `Person`, no characteristic `shoesize` is defined yet, a *smart acting* ActiveTM object tries to find a name or an occurrence with an identifier matching “shoesize”. This works through Ruby’s `method_missing` which handles calls to non-existing methods. Analogously to the getter, using the not-yet-defined setter creates an occurrence¹⁹:

```
p.shoesize = 38 # creates occurrence,
               # type "shoesize", datatype "xsd:integer"
p.shoesize # returns 38
```

¹⁹ Given a string, also a topic name could be created. This depends on the concrete implementation of this function.

Upon success, a set of getters and setters may be created to minimize the overhead of search names and occurrences another time.

Additionally, also the topic names of name types and occurrence types could be searched to find objects corresponding to the method name of the undefined methods.

By its nature, this kind of ontology introspection is highly experimental and may be suitable for programmatically exploring a topic map but not for productive use.

7 Conclusion and Outlook

ActiveTM augments the possibilities of Ruby Topic Maps in a productivity-enhancing way. It enables usage of domain-specific access while not constraining the generic Topic Maps API. ActiveTMML can be used to define ontologies and generate code, code snippets, and ontology documentation. The explicit definitions clearly define the resulting code and thus provide a predictable behavior independent of the data in the topic map. This enables productive usage of ActiveTMML definitions and ActiveTM classes. The intersection of the conceptual design between TMCL and ActiveTMML suggests quite a lot synergies and should be further exploited.

The introspection is rather experimental and not suitable for productive environments. Changes in the data can result in completely different code generated and render the application unusable. Still, it may be interesting to experiment with the introspection, to develop more sophisticated algorithms to look into the topic maps or interpret commonly used modeling patterns to aid writing code for productive usage.

References

- [1] ISO/IEC IS 13250-2:2006: Information Technology – Document Description and Processing Languages – Topic Maps – Data Model. International Organization for Standardization, Geneva, Switzerland.
<http://www.isotopicmaps.org/sam/sam-model/>
- [2] BOCK, B.: Ruby Topic Maps, In: MAICHER, L., GARSHOL, L.M.: *Scaling Topic Maps*. LNAI 4999, Springer, Berlin (2008)
- [3] AHMED, K., GARSHOL, L.M., GRONMO, G.O., HEUER, L., LISCHKE, S., MOORE, G.: *Common Topic Map Application Programming Interface, 2004*
<http://www.tmapl.org/>

- [4] HOLJE, E., SCHMIDT, J.: *PHPTMAPI*.
<http://phptmapi.sf.net/>, 2006-11-10
- [5] HEUER, L.: *Semagia Mappa - The Python Topic Maps engine*. 19 April 2008.
<http://code.google.com/p/mappa/>
- [6] FOWLER, M.: *Domain Specific Language*.
<http://www.martinfowler.com/bliki/DomainSpecificLanguage.html>,
13 February 2004
- [7] FOWLER, M.: *Language Workbenches: The Killer-App for Domain Specific Languages?*
<http://www.martinfowler.com/articles/languageWorkbench.html>,
12 June 2005
- [8] BOGACHEV, D.: *COBOL and Topic Maps? Open Session at TMRA 2007*.
[http://homepage.mac.com/dmitryv/TopicMaps/
TMRA2007/CobolAndTMs.pdf](http://homepage.mac.com/dmitryv/TopicMaps/TMRA2007/CobolAndTMs.pdf). 2007-10
- [9] BOGACHEV, D.: *Subject-centric programming language or what was good about COBOL*.
Blogentry. [http://subjectcentric.com/post/Subject-
centric_programming_language_or_what_was_good_about_COBOL](http://subjectcentric.com/post/Subject-centric_programming_language_or_what_was_good_about_COBOL).
23. October 2007
- [10] BOGACHEV, D.: *Authoring topic maps using Ruby-based DSL: CTM, the way I like it*.
[http://subjectcentric.com/post/Authoring_topic_maps_using_Ruby_ba
sed_DS%L_CTM_the_way_I_like_it](http://subjectcentric.com/post/Authoring_topic_maps_using_Ruby_based_DS%L_CTM_the_way_I_like_it), 28 February 2008
- [11] ISO/IEC Draft 13250-6:2007: Information Technology – Document Description and
Processing Languages – Topic Maps – Compact Syntax, 2007-11-16. International
Organization for Standardization, Geneva, Switzerland.
<http://www.isotopicmaps.org/ctm/>
- [12] ISO/IEC FCD 19756: *Information Technology – Document Description and Processing
Languages – Topic Maps – Constraint Language*. International Organization for
Standardization, Geneva, Switzerland, 2008-08-12
<http://www.isotopicmaps.org/tmcl/>
- [13] BURBECK, S.: *Applications Programming in Smalltalk-80(TM): How to use Model-View-
Controller (MVC)* 1987.
- [14] KILOV, H.: *From semantic to object-oriented data modeling* Bell Commun. Res.,
Morristown, NJ. In: *Systems Integration '90., Proceedings of the First International
Conference on Systems Integration*, 1990 pages: 385–393 ISBN 0818690275
- [15] FOWLER, M., RICE, D.: *Patterns of Enterprise Application Architecture*. Addison-Wesley,
2003. – ISBN 0321127420
- [16] MOORE, G., AHMED, K., BRODIE, A.: Topic Map Objects. In: *Leveraging the Semantics of
Topic Maps*, 2006
- [17] OMG.ORG: Unified Modeling Language Specification.
<http://www.omg.org/technology/documents/formal/uml.html>

Streaming Topic Maps API

Lars Heuer

Semagia
heuer@semagia.com

Abstract. This paper introduces a new, event-based API to create topic maps. It is independent of particular Topic Maps processors and enables developers to convert any resource into a topic map representation with minimal effort.

1 Introduction

The Topic Maps API (TMAPI [12]) is the de facto standard to create and manipulate topic maps in a Topic Maps processor independent way. Even if this API is supported by several Open Source and commercial implementations, it requires some learning effort and is (at least in the upcoming version 2.0) very strict regarding Topic Maps Data Model (TMDM [4]) constraints. The observance of these constraints requires some work if a resource (not necessarily a serialized topic map) should be transformed into a Topic Maps representation. Even worse, this work has to be done for every transformation which leads into repetition of common tasks.

This paper proposes an event-based API which aims to ease transformations of arbitrary resources into a Topic Maps representation with minimal effort. While TMAPI can be seen as Document Object Model (DOM [13]) for Topic Maps, the event-driven API is aligned to the Simple API for XML (SAX [10]) and provides similar advantages over TMAPI like SAX over DOM.

The initial API was implemented in Java but it has been ported to the programming languages Python ([9]) and PHP ([8]). Even if this paper compares the event-API against TMAPI, the proposed API has no relationship to the common Topic Maps API, it serves just as an example to emphasize the difference between a push API and a DOM-alike interface.

2 Design Overview

SAX has proven to be effective and due to its simplicity and elegance it has been ported to several platforms. A parser sends notifications to a handler which processes them accordingly. The parsing process is unidirectional: The parser simply pushes the events to a handler and just forgets about them afterwards. The handler contains the logic to interpret the events and to create something meaningful from it.

The event-based API for Topic Maps does something similar: A parser notifies a handler about information items and their properties. The parser does not take care how these events are processed and ignores merging operations, it simply fires a stream of events. Like for SAX, the pivot for the Streaming Topic Maps API is a handler, called *IMapHandler*. Figure 1 shows an excerpt of callback methods which are used by a parser to send notifications.

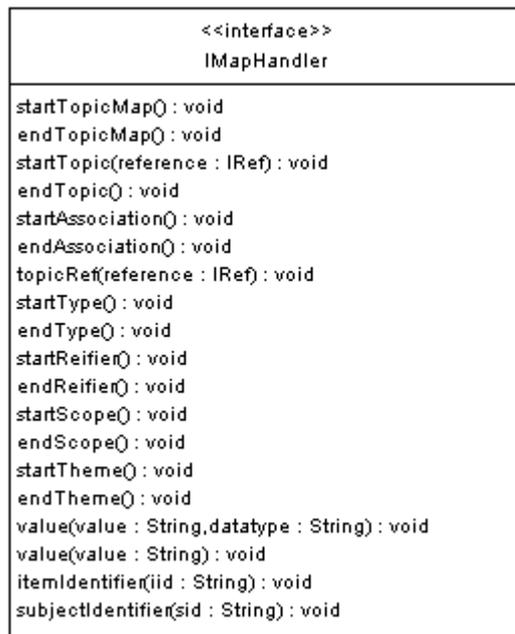


Fig.1. Abbreviated IMapHandler

Altogether, the *IMapHandler* provides 32 callback methods which cover all facets of TMDM. Each information item is introduced with a *start* event and

terminated by an *end* event. The *IRef* interface represents either a subject identifier, a subject locator, or an item identifier and is used to avoid callbacks like *startTopicWithItemIdentifier* or *topicRefWithItemIdentifier*. The *IRef* interface represents always an absolute IRI ([3]). Events like *startType()* and *itemIdentifier(String)* are context sensitive: The IMapHandler implementation has to remember which information item is in the focus and process the notification accordingly.

Given the following CTM [6] fragment:

```
John-Lennon
  - "John Lennon"
.
member-of (member: John-Lennon, group: The-Beatles)
```

A parser reading the fragment will generate a sequence of events like the following (for readability the IRIs are abbreviated to an identifier; the complete IRI for John-Lennon would look like `http://www.example.org/beatles-map.ctm#John-Lennon`):

- *startTopicMap* which is always the first event.
- *startTopic* with an item identifier reference to John-Lennon.
- *startName* to notify the handler that all subsequent events (like *itemIdentifier*) must be interpreted "relative" to the topic name.
- *value* with the string John Lennon
- *endName* to indicate that the name has been parsed
- *endTopic* to indicate the end of the topic John-Lennon
- *startAssociation* to indicate that an association is parsed
- *startType* which indicates that the subsequent topic or topic reference should be interpreted as association type.
- *topicRef* with a item identifier reference member-of
- *endType* precluding the end of the association type processing
- *startRole* Start of a role which participates in the association
- *startType* Role type processing
- *topicRef* with the item identifier reference member
- *endType* indicating the end of role type processing
- *startPlayer* to indicate that the player is processed
- *topicRef* with the item identifier reference John-Lennon
- *endRole*
- ... (processing the other role is omitted)

- *endAssociation*
- *endTopicMap* which is always the last event reported to a handler

As seen, the parser only notifies the handler about events. The parser provides no Topic Maps-related logic, i.e. detecting that a topic with the item identifier `John-Lennon` has already been read and that the latter reference to `John-Lennon` must cause a merge.

It should be emphasized here, that the *IMapHandler* implementation must be done only once for each API (i.e. TMAPI) and that it is relatively easy to implement. To retain the analogy to SAX, the event-based API inverts the responsibility of transforming a resource into a data structure insofar as the parser is variable part here, while the developer has to implement the handler logic in SAX.

The event-driven API allows nesting of events provided that the events cannot be misinterpreted by the *IMapHandler* and that each *start* event has an analogous *end* event. A parser may send a *startTopic* event followed by another *startTopic* event. Due to this flexibility it reduces the development effort for syntaxes which allow nested information items (like CTM). The possibility to interleave events is also the reason why the handler provides callback methods like *startReifier*: After such an event either a complete topic or a reference to a topic may be reported.

As indicated in the introduction, the API is not limited to parse serialized topic maps: A parser can read any imaginable resource and convert it into a Topic Maps representation. An e-mail, an Excel table, the result of a database query, nearly everything is convertible into a sequence of events.

Further, the *IMapHandlers* are stackable: A handler can operate upon another handler and filter the events. If all item identifiers should be omitted, the *itemIdentifier* event is simply consumed by the first handler and not passed on the underlying handler.

3 Push vs. Pull API

Recently, APIs like the Streaming API for XML (StAX [7]) which offer a pulling interface have become popular. These APIs provide typically an iterator over the XML information set and the user is responsible to extract the information which is useful for a particular application. This approach works very well for XML but it seems to be inappropriate for different Topic Maps notations; each Topic Maps syntax provides its own set of features, and creating a common interface for all notations would be a step away from the simplicity of the *IMapHandler*.

Due to the stackability of the *IMapHandler* interface it should be easy to implement filters which omit certain kinds of information items. Based on the fact that topic maps are usually imported as a whole, the pull-approach seems to have no advantage over the API presented in this paper. Pulling information from a resource implies knowledge about the concrete data structure of the resource while pushing the events is much more lightweight and easier to deploy, especially for arbitrary resources.

4 Serializing Topic Maps

SAX can be used to serialize a data structure into a sequence of events which result into XML elements and attributes. Even if the *IMapHandler* can be used like the SAX counterpart, it seems to be more difficult since Topic Maps provides different serialization formats. A syntax like CTM offers nested Topic Maps constructs, while XTM disallows such a structure. How should a Topic Maps processor report the information items?

Due to its outstanding position, the XTM format seems to be the ideal candidate for a pattern in which sequence events are reported, but taking that pattern would lead to a degenerated CTM serialization. The *IMapHandler* for CTM demands other requirements on the order of events as a XTM serializer. Given the fact that there is no common superset on how a topic map should be serialized, the effort to use the *IMapHandler* to serialize Topic Maps has been abandoned.

5 Conclusions and Further Work

The event-driven API has proven to be deployed easily. The pivot *IMapHandler* has to be adapted for the Topic Maps processor-specific API, but it can be used for all kind of parsers.

The Open Source Topic Maps processor tinyTiM [11] uses the event-driven API to import all kind of Topic Maps syntaxes. The author of this paper has implemented a framework, called MIO [2] around the API which allows the discovery of Topic Maps deserializers which use the *IMapHandler*. While porting tinyTiM from TMAPI 1.0 to 2.0, the event-driven API in conjunction with the MIO framework has proven to be simple, since parser logic was not affected: The deserializers use the *IMapHandler* and the TMAPI changes are transparent for them. A deserializer which works for the predecessor also works for the latest version of tinyTiM.

Further, a binary representation of the event sequences, called Binary Topic Maps (BTM [1]), has been implemented. This binary format reduces the storage requirements for serialized topic maps considerable and should be convenient to send the events over a network.

The API has been introduced to the TMAPI project and gained a lot of positive feedback. Unfortunately due to lack of human resources it was not yet integrated into the project. Even though it would be desirable to adopt the event-driven approach widely.

It would be interesting to elaborate the possibility to implement the API on top of a RDF store but this was not yet done. The opposite (a RDF parser which sends events to the handler) is currently implemented.

References

1. L. Heuer. Binary Topic Maps (BTM). <http://www.semagia.com/tr/btm/1.0/>
2. L. Heuer. Topic Maps I/O (MIO). <http://mio.semagia.com/>
3. Internet Standards Track Specification. Internationalized Resource Identifiers (IRIs). <http://www.ietf.org/rfc/rfc3987.txt>
4. ISO/IEC. IS 13250-2:2006: Information Technology — Document Description and Processing Languages — Topic Maps — Data Model. Technical report, International Organization for Standardization, Geneva, Switzerland., 2006. <http://www.isotopicmaps.org/sam/sam-model/2006-06-18/>
5. ISO/IEC. IS 13250-3:2006: Information Technology — Document Description and Processing Languages — Topic Maps — XML Syntax. Technical report, International Organization for Standardization, Geneva, Switzerland., 2006. <http://www.isotopicmaps.org/sam/sam-xtm/2006-06-19/>
6. ISO/IEC. FCD 13250-2: Information Technology — Document Description and Processing Languages—Topic Maps—Compact Syntax (CTM) 2008-05-15. Technical report, International Organization for Standardization, Geneva, Switzerland., 2008. <http://www.isotopicmaps.org/ctm/ctm.html>
7. Java Community Process JSR 173. Streaming API for XML (StAX). <http://jcp.org/en/jsr/detail?id=173/>
8. PHP project. PHP. <http://www.php.net/>
9. Python project. Python. <http://www.python.org/>
10. SAX project. Simple API for XML (SAX). <http://www.saxproject.org/>
11. tinyTiM project. tinyTiM. <http://sourceforge.net/projects/tinytim>
12. TMAPI project. Topic Maps API. <http://www.tmapi.org/>
13. W3C. Document Object Model (DOM). <http://www.w3.org/DOM/>

Protocol for the Syndication of Semantic Descriptions

Graham Moore¹, Marc Wilhelm Küster²

¹Networked Planet
graham.moore@networkedplanet.com

²FH Worms – University of Applied Sciences
kuester@fh-worms.de

Abstract. This document describes a protocol to be used for the exchange of semantic descriptions. The protocol defines how a web service can publish a series of web accessible feeds that describe snapshots and changes to a collection of semantic descriptions. This protocol also defines how a client should process the feeds provided by the service such that a local store is in sync. A client can synchronize with more than one server to act as an aggregator for semantic descriptions.

Overview

This document describes a protocol to be used for the exchange of semantic descriptions. The protocol defines how a web service can publish a series of web accessible feeds that describe snapshots and changes to a collection of semantic descriptions. This protocol also defines how a client should process the feeds provided by the service such that a local store is in sync. A client can synchronize with more than one server to act as an aggregator for semantic descriptions.

The current version of the specification, which is still subject to revision in the standardization process, is available at http://www.egovpt.org/fg/CWA_Part_1b

Scope

This document specifies the underlying syndication protocol for the exchange of information about semantic descriptions. The protocol conforms to the Atom Syndication Format and the TMDM syntax and model. It defines several layers

Maicher, L.; Garshol, L. M. (eds.): *Subject-centric computing. Fourth International Conference on Topic Maps Research and Applications, TMRA 2008, Leipzig, Germany, October 16-17, 2008, Revised Selected Papers.* (Leipziger Beiträge zur Informatik: XII) - ISBN 978-3-941152-05-2

of syndication feeds that must be provided by a conforming application. Finally it defines algorithms for the provision and processing of the different feeds on the server and on the client.

Concepts

This protocol defines how a server can produce a number of Atom feeds that describe either a list of topic maps that the server manages, a list of snapshots of a given topic map or a list of topic map fragments. Each fragment is created because it is the new representation of a topic that has changed.

A client that wishes to maintain a topic map in sync with one held on the server first fetches the most recent snapshot for the required map and stores it locally. It then subscribes to the feed of 'changes' for that map. This feed lists topicmap fragments. A fragment is created and an entry added to the feed when a topic is updated, added or deleted from the topic map. The client then updates its local topicmap with the new topic representation.

This protocol defines the structure of the feeds published by the server and how a client should interpret and process these feeds.

Note: the notion of a client and server is solely defined by the responsibilities of each. Thus a given machine can act both as client and server (peer-to-peer scenario) or restrict itself to exactly one of the roles (publish-subscribe scenario).

Protocol

Terminology

Server Node: A node hosting both feed and data services that allow a client to understand the state of the topic map being managed over time.

Client Node: A node that subscribes to one or more server nodes and implements the update semantics defined in this protocol.

Server Contract

A server node is responsible for providing information about the state of the topic map(s) it is managing. It provides a number of feeds that allow clients to see

which aspects of the map has changed over time and data services that allow a client to fetch representations of the topic map or individual topic instances in order to update a client environment.

Feeds & Data Services

A compliant server will provide the following Atom 1.0 feeds, fragment data services and snapshot data services.

Topic Maps Feed - a list of all the topic maps being managed by the server.

Example Service URL: [server]/topicmaps

Example Invocation:

```
Server: http://tmshare.networkedplanet.com
GET /topicmaps
```

The Atom payload of the topic maps feed contains an entry for each topic map. Each entry has a link to an Atom feed for the specified topic map.

Example response body:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Topic Maps managed by
    tmshare.networkedplanet.com</title>
  <link href="http://tmshare.networkedplanet.com/" />
  <updated>2008-12-13T18:30:02Z</updated>

  <author>
    <name>TMShare Server</name>
  </author>

  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
  <!-- topic map entry -->
  <entry>
    <title>eGovernment Resources Topic Map</title>
    <!-- a link to the atom feed that has entries linking to
    the snapshots and fragments feed for this map -->
    <link rel="topicmapfeed" type="application/atom+xml"
      href="http://tmshare.networkedplanet.com/topicmaps/
      egov"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2008-12-13T18:30:02Z</updated>
    <summary>A topic map that contains the classification of
      eGovernment services.</summary>
  </entry>
```

```

    <!-- an entry follows for each topic map being exposed.
        ...
    -->
</feed>

```

Topic Map Feed - a feed for a given topic map that provides exactly two entries, one linking to a snapshots feed and one to a fragments feed.

Example Service URL: [server]/topicmaps/egov

Example Invocation:

```

Server: http://psi.egovpt.org
GET /topicmaps/egov

```

The Atom payload of the response contains the two entries, one that links to a feed with all snapshots of the topic map (`rel`-attribute of the link is `snapshotfeed`), and another one that links to an Atom feed that lists changes to the topic map (`rel`-attribute of the link is `fragmentfeed`). Optionally, both links can be duplicated in the entries with `rel`-attributes that have the value `alternate`. This helps Atom feed readers to correctly display the links.

```

<a:feed xmlns:a="http://www.w3.org/2005/Atom">
  <a:title>eGov TM</a:title>
  <a:updated>2008-09-26T11:13:40-01:00</a:updated>
  <a:subtitle>Feeds around the eGov TM</a:subtitle>
  <a:id>http://http://psi.egovpt.org/feeds/testtm/</a:id>
  <a:author>
    <a:name>Isidor</a:name>
  </a:author>
  <a:link href="http://http://psi.egovpt.org/feeds/testtm/"
rel="self"/>
  <a:entry>
    <a:title>eGov TM: Fragments</a:title>
    <a:id>http://http://psi.egovpt.org/testtm/fragments/
      </a:id>
    <a:updated>2008-09-11T17:58:39-01:00</a:updated>
    <a:author>
      <a:name>Isidor</a:name>
    </a:author>
    <a:link
      href="http://http://psi.egovpt.org/testtm/fragments/"
      rel="alternate" type="application/atom+xml"/>
    <a:link
      href="http://http://psi.egovpt.org/testtm/fragments/"
      rel="fragmentfeed" type="application/atom+xml"/>
  </a:entry>
  <a:entry>
    <a:title>eGov TM: Snapshots</a:title>

```

```

<a:id>http://http://psi.egovpt.org/testtm/snapshots/
  </a:id>
<a:updated>2008-09-11T17:58:39-01:00</a:updated>
<a:author>
  <a:name>Isidor</a:name>
</a:author>
<a:link
  href="http://http://psi.egovpt.org/testtm/snapshots/"
  rel="alternate" type="application/atom+xml"/>
<a:link
  href="http://http://psi.egovpt.org/testtm/snapshots/"
  rel="snapshotfeed" type="application/atom+xml"/>
</a:entry>
</a:feed>

```

Snapshot Feed - a list of all the representations of a given topic map over time. At present, XTM 1.0 representations are supported.

Example Service URL: [server]/topicmaps/egov/snapshots

Example Invocation:

```

<?xml version="1.0"?>
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:tmshare="http://www.egovpt.org/tmshare">
  <title>The Snapshots of the eGovernment
    Resources Topic Map</title>
  <subtitle>A list of all XTM representations of
    this map</subtitle>
  <author>
    <name>TMShare Server</name>
  </author>
  <updated>2008-07-17T12:15:07.020071Z</updated>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <tmshare:ServerSrcLocatorPrefix>http://psi.networkedplanet.
  com/</tmshare:ServerSrcLocatorPrefix>

  <!-- a link to the feed -->
  <link rel="self"
    href="http://tmshare.networkedplanet.com/topicmaps/
    egov/snapshots"/>
  <entry>
    <title>Snapshot 2008-07-17</title>
    <updated>2008-07-17T14:04:42.205299Z</updated>
    <!-- a link to the XTM 1.0 snapshot -->
    <link rel="topicmapdata" type="application/xtm1+xml"
      href="http://tmshare.networkedplanet.com/topicmaps
      /egov/shapshots/60a76c80-d300-11d9-b93C-
      0003939e0af6"/>
    <id>60a76c80-d300-11d9-b93C-0003939e0af6</id>

```

```

</entry>
<!-- an entry follows for each XTM snapshot being exposed.
...
-->
</feed>

```

Topic Map Fragments Feed - a list of topic map fragments that indicate changes for a given topic map over a period of time.

Example Service URL: [server]/topicmaps/egov/fragments

Example Invocation:

```

Server: http://tmshare.networkedplanet.com
GET /topicmaps/egov/fragments

```

Example response body:

The Atom payload contains an entry for each fragment. Each entry contains one link to the fragment and the updated element contains the time at which the fragment was created. In addition to the standard Atom elements this protocol introduces two new elements.

The new elements are:

<ServerSourceLocatorPrefix>: Indicates to a client the prefix to use to locate topic properties that should be removed when updating a topic. This element should occur once as a child element of the <feed> and before the first entry. (See the fragment update algorithm below for more information).

And

<TopicSI>: Indicates to a client which topic is being updated from all those present in the fragment. This element **MUST** occur once as a child element of each <entry>. (See the fragment update algorithm below for more information).

```

<?xml version="1.0"?>
<feed xmlns="http://www.w3.org/2005/Atom"
xmlns:tmshare="http://www.egovpt.org/tmshare">
  <title>Change fragments from the eGovernment Resources
    Topic Map</title>
  <author>
    <name>TMShare Server</name>
  </author>
  <updated>2008-07-17T15:47:17.062211Z</updated>
  <id>28C5DBD8-652A-4617-8C4A-C0FFC49B4475</id>
  <!-- The serversrclocatorprefix is used by a client
    to know the providence of topic map constructs. -->
  <tmshare:ServerSrcLocatorPrefix>http://psi.networkedplanet.

```

```

com/</tmshare:ServerSrcLocatorPrefix>
  <link rel="self"
        href="http://tmshare.networkedplanet.com/
            topicmaps/egov/fragments"/>
  <entry>
    <!-- Best practice: the topic display name or the PSI
should be used for the entry title -->
    <title>ISO 19115:2003 Geographic Information -
        Metadata</title>
    <!-- the published date and time of the fragment -->
    <updated>2008-07-17T15:55:21.971145Z</updated>
    <!-- the id value is some unique value -->
    <id>69CD5264-DB78-49c1-A7E4-04EECF0AA85</id>
    <link rel="topicmapdata" type="text/xml+xml"
          href="http://tmshare.networkedplanet.com/topicmaps/
            egov/fragments/69CD5264-DB78-49c1-A7E4-
            04EECF0AA85"/>

    <tmshare:TopicSI>http://psi.egovpt.org/standard/ISO+19115
%3A+Geographic+Information+-+Metadata</tmshare:TopicSI>
  </entry>
  <!-- an entry follows for each fragment being exposed
        ...
-->
</feed>

```

Topic Map Fragment Data Service - a service that returns a specified topic map fragment.

Example Service URL:

```
[server]/topicmaps/egov/fragments/fragment-for-topic-1
```

Example Invocation:

```
Server: http://tmshare.networkedplanet.com
GET /topicmaps/egov/fragments/fragment-for-topic-1
```

Response structure:

A topic map fragment representation is a valid XTM 1.0 XML document. A fragment is created in the context of exactly ONE topic. The following algorithm should be applied when generating a fragment for given topic:

- Let 'export' mean to create an XTM representation of the TMDM construct
- Let T be the topic being exported.
- export T including ALL topicnames, identifiers, and occurrences.

- for each topicname in T export a topic stub for each name type (if it exists)
- for each topicname in T export a topic stub for each scope topic (if it exists)
- for each occurrence in T export a topic stub for the occurrence type (if it exists)
- for each occurrence in T export a topic stub for each scope topic (if it exists)
- for each association A in which T plays a role export the association
- for each association A export a topic stub for the association type
- for each association A export a topic stub for each topic scope topic
- for each role R in A export a topic stub for the role type and one for the role player UNLESS the role player is T

For each stub topic exported (the following minimum must be exported)

- export ALL of the topic's identifiers

ALL topics (stub or not) MUST have at least one Subject Identifier.

An server may choose to export more information in the fragment, what is described here is the minimum required.

Client Responsibilities

There are two aspects to client behaviour. The first is consumption of the feeds provided by the service the second is the updating of the local map based on the fragments it retrieves.

A Clean start

When a client first wants to sync with a server it can use the feeds provided to locate the topic map of interest, retrieve the full XTM topic map representation and merge it into the local topic map it is managing.

A Clean Replacement

If a client has a local topic map that contains topic map data from more than one server and wants to fetch and update the latest full topic map from ONE source then it MUST do the following. Apply the delete topic algorithm from below, but apply it to the entire topic map. Then proceed in terms of 'A Clean Start', by fetching the topic map and merging it in.

A partial update

Clients wishing to update their local topic map as new changes occur on the server, should process the changes feed for the appropriate topic map. The client MUST record the date and time that it last updated its local copy and then find all Atom entries that have an updated value after that time. For each of these, in time order of most distant to most recent it should apply the following update algorithm.

The Topic Map Fragment Update Algorithm_

- Let SP be the value of the `ServerSourceLocatorPrefix` element in the Atom feed F
- Let SI be the value of `TopicSI` element in Atom entry E
- feed F contains E
- entry E references topic fragment TF
- Let LTM be the local topic map
- Let T be the topic in LTM that has a `subjectidentifier` that matches SI
- For all names, occurrences and associations in which T plays a role, TMC
 - Delete all `srcLocators` of TMC that begin with SP
 - If the count of `srclocators` on TMC = 0 then delete TMC
- Merge in the fragment TF using SP as the base all generated source locators.

Note: To delete a topic an empty topic is published.

Note: The understanding is that each name, occurrence and association created or modified during the update will in its internal, TMDM-conformant

representation have or get item identifiers that act as source locators and start with the `ServerSourceLocatorPrefix`.

References

1. XTM 1.0
<http://www.topicmaps.org/xtm/index.html>
2. XTM 2.0
<http://www.isotopicmaps.org/sam/sam-xtm/>
3. RFC 2616 HTTP 1.1
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>
4. XML 1.0
<http://www.w3.org/TR/REC-xml/>
5. RFC 4287 Atom Syndication Format 1.0
<http://www.atompub.org/rfc4287.html>

Living Topic Maps

Creating Web Presentation for Observatory and Planetarium with Topic Maps

Martina Husáková and Kamila Olševiřová

Faculty of Informatics and Management, University of Hradec Králové,
Hradecká 1227, Hradec Králové, Czech Republic

{martina.husakova.2, kamila.olsevicova}@uhk.cz

Abstract. The aim of the Topic Maps application for astronomers and visitors of Observatory and Planetarium in Hradec Králové is to help them to search resources related to astronomy. The Topic Maps document can be immediately consulted during presentations and courses for public and can be reused for creation of web presentation of the Observatory. In the paper the process of the application development is summarized.

Keywords: Topic Maps, TM4L, Ontopia Navigator Framework, astronomy

1 Introduction

Semantic web is understood as the environment where software agents browse web page from one to another and perform sophisticated tasks on behalf of human users. To enable this, it is necessary to develop new technologies for encoding the meaning and context of each piece of information presented on web sites. An international standard ISO/IEC 13250 Topic Maps [6] is intended on realizing ideas of the semantic web.

The Topic Maps standard uses knowledge representation schema – Topic Map. We perceive this structure as some Topic Maps document written in certain syntax (XTM [4], LTM [2], etc.). It is composed of three basic elements – topic, association and occurrence. Having these elements, it is possible to create metadata layer describing digital sources of different types to facilitate access to them.

In this paper we introduce an application of Topic Maps approach that is related to e-learning, searching services and information delivery. We focus on creating

web presentation for needs of the Observatory and Planetarium in Hradec Králové.

An unsuitable solution for organizing and convenient searching relevant terms connected with astronomy domain is used at the Observatory. Our Topic Maps document will help to access web pages related to astronomy and to ensure better navigation.

2 Original application used at the Observatory

At the Observatory and Planetarium in Hradec Králové, presentations and courses about astronomy are organized. For this purpose – and also for their own research – astronomers need to manage huge amount of digital information and knowledge resources. Astronomers were used to work with file-manager-like application written in Tcl/Tk. It enables browsing local digital repository during the lectures for public through the system of menus (in Czech), with main categories such as *Presentations, Animations, Pictures – physics* etc. E.g. the list of *Pictures - physics* includes the titles such as *Meteorological radar, Meteors - Perseids, Milky Way - structure, Molecular clouds* etc., see Fig. 1 [5].

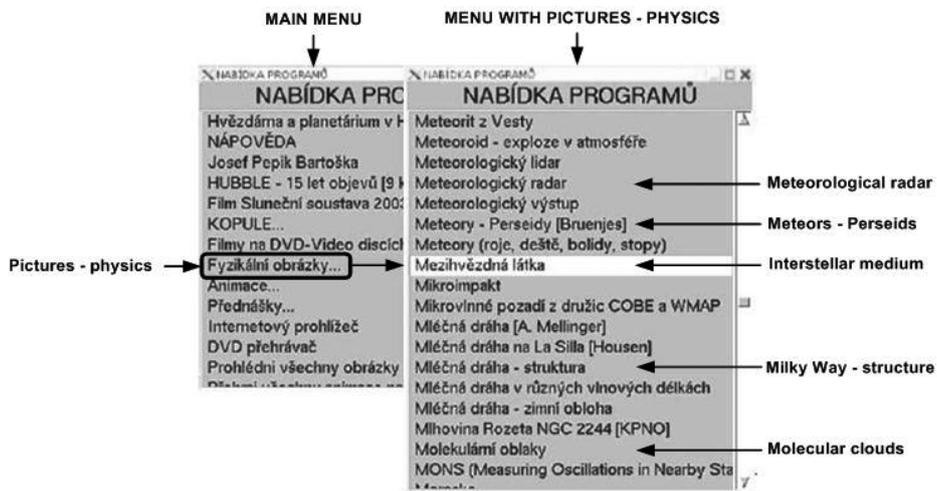


Fig. 1. Original solution in Tcl/Tk

The items inside main categories are ordered randomly. No hierarchies and no relations between concepts are defined, therefore it is not possible to differentiate

between general and special terms. The system is not user friendly, items are ordered alphabetically and in certain cases the lecturer has to remember the content of menus to be able to search it quickly.

Our objective is to help astronomers by designing and implementing the Topic Maps application. It should simplify accessing relevant resources during lectures and help to better organize the content of the Observatory website.

3 Topic Maps document creation

Development of the Topic Maps document was realized in diploma thesis [5]. The key parts of this thesis were: analysis of the original current application and design development of the pilot project based on the Topic Maps standard.

The Topic Maps document was realized in cooperation with the domain expert, who is its future user (being lecturer at the Observatory). The participation of the expert was necessary to ensure the correctness of the final structure of the Topic Maps document. Following requirements for the Topic Maps document were defined, see Tab. 1.

Table 1. Requirements for the Topic Maps document

User's requirements	Priority	Comment
Capturing the ontology of astronomical terms	high	
Topics implementation	high	
Implementation of associations among topics to enable navigation	high	This was expected to be the main benefit for users.
External occurrences implementation	medium	Lecturers often work with hyperlinks from Wikipedia.
Defining internal occurrences	medium	Internal definitions are not necessary.
Defining topic names in different languages	high	At least in English and Czech

Following requirements related to the Topic Maps editor were defined, because the updating and the maintenance of the Topic Maps document will be performed by users (lecturers and astronomers):

- The management of the Topic Maps document has to be intuitive, because astronomers are not IT specialists and do not want to study programming languages.

- Visualization and searching tools for the Topic Maps document content are required.
- Periodical automated testing of functioning of web links, stored as external occurrences in Topic Maps document, is important.
- Simple installation and maintenance, Windows version is preferred, low costs on future enhancing the application.

The first version of the Topic Maps document was created in editor Ontopoly, part of Ontopia Knowledge Suite (OKS) package [7], that fulfills most of requirements. The Topic Maps document was realized in iterative way. Requirements specification, design, implementation, integration, testing and debugging, were repeated several times to reflect comments and recommendations given by expert. During the design phase, elements for the Topic Maps document (topics, associations, etc.) were collected and organized into hierarchies. We could find out if any inconsistencies exist in the Topic Maps document with the aid of validation function that is offered by editor Ontopoly.

After consultations with expert was found out that the Topic Maps technology is suitable. The Topic Maps document could be used with OKS Samplers [8] for the pilot version. Experiences and results ensured us about practical usage of the Topic Maps document. Further requirements for the web application were specified, see Tab. 2.

Table 2. Requirements for the web application

User's requirements	Priority	Comment
Intuitive navigation	high	
Clear organization of web pages	high	
Possibility to search Topic Maps elements	high	
Possibility to visualize Topic Maps document in expandable graph structure	high	Mainly for educational purposes
Possibility to edit elements of the Topic Maps document from web-based forms	medium	
Online presentation of the Topic Maps document	high	In the initial phase of the development, application is tested offline.

The Topic Maps document was created using Ontopoly editor and it was clear that the web-based application will be developed with next component of OKS

environment – scripting language based on XML - Ontopia Navigator Framework (ONF) [9].

Firstly experiments with ONF and the Topic Maps document were performed. It was seen that ONF is quite easy to use, but one problem occurred. If we had created web pages with ONF, we used various special tags. These tags can contain identifiers of the Topic Maps elements. If we had modified the Topic Maps document in Ontopoly, identification values changed. This was very unpleasant observation after long effort. Our next steps were focused on searching more suitable tool, without problems with identifiers and also eligible for users.

4 Comparison of Topic Maps tools

Following three editors of Topic Maps documents creation were investigated with respect to the given web application developers' and users' requirements:

- Wandora [12], knowledge management solution based on Topic Maps principles, which supports export into HTML and therefore can be used for web presentation creation,
- TMTab plug-in [11] for Protégé ontology editor that allows exporting the ontology into XTM syntax,
- TM4L editor (Topic Maps 4 E-learning) [1], an editor of Topic Maps documents, developed for educational purposes.

Experiments were made with LTM syntax [2] written in NotePad too, but this manual approach would be uncomfortable for the customer. Tab. 3 summarizes properties of tools. Editor Ontopoly is also mentioned for comparison with others tools.

Finally TM4L environment was chosen because of properties that are compliant much more than editor Ontopoly. It bears on among other things steady identification values of Topic Maps elements if you modify Topic Maps document, see more details in Tab. 3.

Activities on developing the Topic Maps document did not finish. It was found out that original Topic Maps document could be opened in TM4L editor (ver. 1, 2), but taxonomy was not saved. For solving this problem we saw only one way – to develop the Topic Maps document once more in TM4L editor. We chose TM4L editor version 2, because is more sophisticated than version 1: it can visualize Topic Maps structure and supports tolog query language [3], see Tab. 3.

Table 3. Tools for Topic Maps document creation

Requirements	Ontopoly	TMTab	Wandora	TM4L editor
Intuitiveness	yes	no	no	yes
Visualization	yes (Vizigator)	no	yes	yes
Searching topics	yes	yes	yes	yes
Automated testing of web sources	no	no	no	yes
Installation with no sweat	no	yes	yes	yes
OS Windows	yes	yes	yes	yes
Low price	yes	free	free	free
	(OKS Samplers)			

5 Web presentation creation

Experience with TM4L editor and ONF confirms that combination of these tools is good way for creation pilot version of web with Topic Maps approach – TM4L editor for defining Topic Maps document and ONF for designing web presentation based on the Topic Maps document.

Developing web pages with ONF lies in creating JSP documents containing special tags with queries in tolog language [3]. Their purpose is for example: extracting pieces of information and knowledge from the Topic Maps document, their view on the web pages, supply information from the Topic Maps document under some condition etc. It is not necessary to know details about JSP programming for creating simple web pages with ONF, but knowledge about tolog language is indispensable.

Firstly, scope of pilot web-based presentation had to be mentioned. The domain of astronomy – which is extremely complex, and includes knowledge of mathematics, physics, chemistry, history, philosophy etc. – was restricted after consultation with expert. Only concepts related to selected objects of the Solar System were taken into account, and the aim was to define corresponding ontology and to present relevant information and knowledge resources on these objects.

For web-based presentation based on the Topic Maps document, we chose predefined layout and adjusted it for our purposes. The layout of main pages (about particular objects in the Solar System) can be seen on Fig. 2. The topic name is presented in the heading. The left part of the page contains other names, description of the topic (internal occurrence) and links to web resources, i.e. pictures, animations, documents (external occurrences). Navigation menu and list of associations are published on the right side of the page. Hypertext link to the homepage, information about web and contact to the author is entirely up on the right side of the every web page. Sample page is presented on Fig. 3.

We were focused on quality not for quantity in the first stages of web development. It means that we only chose for example *Discoverers* of some astronomical object and we tried hard to ensure right view, encoding and layout particular information on the page. This strategy was realized with Apache Tomcat servlet/JSP Container, that was installed in our personal computer. We used following tags in web pages: *tolog:context*, *tolog:set*, *tolog:foreach*, *tolog:id*, *tolog:out* and *tolog:if*. Explanation of them can be found in [10].

6 Final state of the pilot application

The web presentation contains web pages dedicated to particular objects of the Solar system - *Planets* - with Czech and English names, description of the object, list of related topics, links to internally stored pictures and to external web resources. Czech-English astronomical dictionary, overview of involved Solar System objects, people (famous astronomers), external resources (animations, pictures, text documents) are presented on special pages. For statistics of the Topic Maps document, see Tab 4. All requirements (Tab. 1) were accomplished.

Table 4. Statistics of the Topic Maps document

Element of the Topic Maps document	Count
topic types	18
instances	69
associations	6
external occurrences	143
internal occurrences	67
themes	10
Total TAOs	313

The pilot web presentation was evaluated by the expert from the Observatory. Regarding requirements in Tab. 2, his conclusion was that both navigation scheme and organization of pages are clear and easy to use.

One of the main benefit is possibility to quickly detect association(s) between topics, for example *Sun* and *Solar flare (has activity)*. This application has also some weaknesses. It has not been accessible through the Internet yet. In the first phase of this web project is tolerable because pilot version of this web is testing with OKS samplers. Searching, editing and visualization elements of the Topic Maps document has not been realized too. It means that only 2 from 6 requirements for web application were solved so far. It is clear what steps are going to follow.

7 Conclusion

This paper describes the process of Topic Maps application creation for the purpose of Observatory and Planetarium in Hradec Králové. Two tools were used, TM4L editor for the defining the Topic Maps document, and Ontopia Navigator Framework for creating the web presentation.

The pilot web presentation contains information and links to knowledge resources about selected objects of the Solar System. Next effort will be focused on extending the Topic Maps document in cooperation with domain experts and mainly on defining the procedures of further maintenance and utilization of the application.

The visualization of the Topic Maps structure should be presented on the website of the Observatory. This can be achieved using VizLet applet that is provided with professional version of OKS. Searching relevant information and their optional editing through web-based forms by authorized users is planned too.

Acknowledgement

The research has been supported by the Czech Grant Foundation, Grant No. 402/06/1325 AmIMaDeS. Authors would like to thank to Mr. Miroslav Brož, expert in astronomy from the Public Observatory in Hradec Králové, who helped to define the domain ontology.

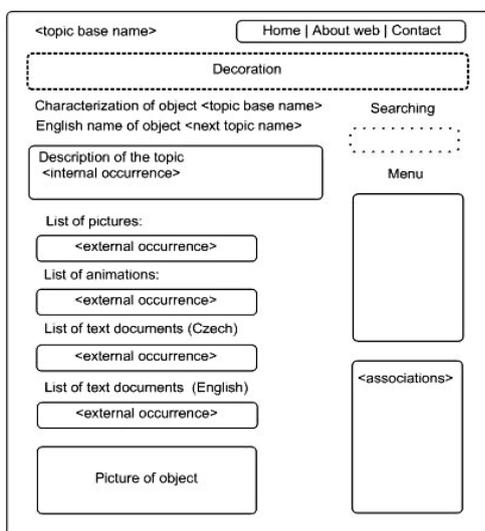


Fig. 2. Layout of web page

Uran By Free CSS Templates home | o webu | kontakt

CHARAKTERIZACE OBJEKTU Uran

Původní název: Georgian Hvězda krále Jiřího
 Anglický název: Uranus

Popis: Jedná se o planetu, která má zhruba dvojnásobnou vzdálenost od Slunce než planeta Saturn. Nedá se pozorovat prostřím okem, ale jen dalekohledy. Má také prstence, ale ty jsou velmi slabé.

Seznam obrázků:

- http://antwrp.gsfc.nasa.gov/apod/image/0108/uranus_vp2.jpg
- <http://private.addcom.de/jsekk/bilder/neptun.jpg>
- <http://www.geocities.com/kzupetic/Uran.jpg>

Seznam webových zdrojů (ČJ):

- http://cs.wikipedia.org/wiki/Uranovy_m%C4%9Bs%C3%ADce
- <http://www.observatory.cz/info/index.php?page=Obloha%20dnes/index.html>
- <http://www.aldebaran.cz/astrofyzika/sunsystem/uran.html>

```

<ul>
<tolog:foreach query=" select $vyskyt from
occurrence(%astro_objekt%, $vyskyt),
scope($vyskyt.x1p7dk5cib-37b)?">
<li><a href="<tolog:out var="vyskyt" />"><tolog:out var="vyskyt"/></a></li>
</tolog:foreach>
</ul>
    
```

Hledání

search

Další odkazy na webu

- [Astronomické objekty](#)
- [Objevitelé](#)
- [Aktivity astronomických objektů](#)
- [Povrchy astronomických objektů](#)
- [Složení atmosféry astronomických objektů](#)
- [Česko-anglický slovník](#)
- [Webové zdroje \(AJ\)](#)

Fig. 3. Sample page of the topic “Uran” with relevant tolog code

References

1. Dicheva, D.: Towards Reusable and Shareable Courseware: Topic Maps-based Digital Libraries, <http://compsci.wssu.edu/iis/nsdl/> (2006)
2. Garshol, L. M.: The Linear Topic Map Notation: Definition and introduction, ver. 1.3, <http://www.ontopia.net/download/ltn.html>
3. Garshol, L. M.: Tolog: A topic map query language, <http://www.ontopia.net/topicmaps/materials/tolog.html>
4. Garshol, L. M., Moore, G. The XML Topic Maps (XTM) Syntax, <http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0398.htm> (2003)
5. Husáková, M.: Integration of information and knowledge sources by means of Topic Maps, University of Hradec Králové, Faculty of Informatics and Management (2007)
6. ISO/IEC 13250 Topic Maps: Document Description and Processing Languages, http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf (2002)
7. Ontopia: Ontopia website, <http://www.ontopia.net/> (2008)
8. Ontopia: OKS Samplers Free Download, <http://www.ontopia.net/download/freedownload.html>
9. Ontopia: Ontopia Navigator Framework, <http://www.ontopia.net/solutions/navigator.html>
10. Ontopia: The Navigator Tag Libraries: Reference Documentation, version 4.0 (2008)
11. TMTab: The Topic Map Tab, <http://www.techquila.com/tmtab/>
12. Wandora: Wandora Features, <http://www.wandora.net/wandora/wiki/index.php?title=Features>

Creating a Topic Maps Based e-Learning System on Introductory Physics

Shu Matsuura¹ and Motomu Naito²

¹ School of High-Technology for Human Welfare, Tokai University, 317 Nishino, Numazu, Shizuoka 410-0395, Japan
shum@wing.ncc.u-tokai.ac.jp

² Knowledge Synergy Inc., 203 Residence Tokorozawa Nibankan, 3-747-4 Kusunokidai, Tokorozawa, Saitama 359-0037, Japan
motom@green.ocn.ne.jp

Abstract. Construction of an introductory physics e-learning system based on Topic Maps is discussed in view of subject-centric design of web-based learning. A pilot system with a visualized Topic Maps portal was created and utilized for students' self-study of university lectures. The aim of this system is to provide a platform where learners can design their study by themselves, and extend their study into information resources on the Internet. The students' response to an inquiry on their impressions of the pilot system suggested that the Topic Maps portal is useful for figuring out the relationships of knowledge, and that navigation for the order of learning materials was required for beginners. Further, an e-Learning Topic Maps system that consists of three main domains, i.e., physics subjects, learning resource types, and learning record types, was created to improve the system extensible in physics related knowledge and in the types of associations between subjects.

Keywords: Topic Maps, e-Learning, introductory physics education

1 Introduction

Web-based e-Learning has gained popularity as a type of learning facilities in many fields. E-Learning can provide learners with an interactive and flexible interface to knowledge resources, and it can be adopted with individual requirements of learners. One of the authors is creating an original e-Learning

system of introductory physics “Everyday Physics e-Learning (EPEL in abbreviation)” [1], which is made open also for public use, and utilizing it as a learning environment for students’ self-study in the university, in addition to the ordinary face-to-face lectures.

The system had been used with the schedule type portal, as shown in fig. 1, in which learning materials were arranged linearly with time, i.e., the dates of lectures, for 4 kinds of courses of dynamics, heat, wave, and electromagnetism. In order to enhance learners’ spaced repetitive learning [2], the system was equipped with an original time-dependent weighted accumulation function to evaluate learners’ drill scores [3]. This learning support system had been mainly used for preparation and review for the face-to-face lectures.

Topic Maps technology has already been applied to the construction of e-Learning system [4, 5]. In the 2006 autumn semester, we introduced a Topic Maps visualization style portal as shown in fig. 2 into the previous schedule type system, in order to represent overall knowledge structures of learning resource the system consisted of, and to stimulate spontaneous self-study over a wider range than that covered by the lectures the students followed. In the maps, button labels showed the names of subjects, and the emphasized arrows pointed base subject of a selected one.

In the renewed system, the equivalent maps of contents were displayed for texts and 4 kinds of drills, i.e., essay drills, multiple-choice drills, calculation drills, and free-style drills. In the drill maps, the colors of button labels were determined by the evaluation values that were calculated by the above weighted accumulation function of drill scores.

From the 2007 spring semester, the system was changed to use completely the Topic Maps visualization style, and the previous schedule type portal was removed. The aim of this change is to make the system be a platform where learners are able to design their way of study by themselves, even independent upon lecture schedules, and the teacher can evaluate students’ individual study.

From the 2007 autumn semester, the authors are reconsidering the construction of the system from the viewpoint of subject-centricity, and are trying to make it to be a platform where students start their study and extend into broad knowledge resource of the Internet.

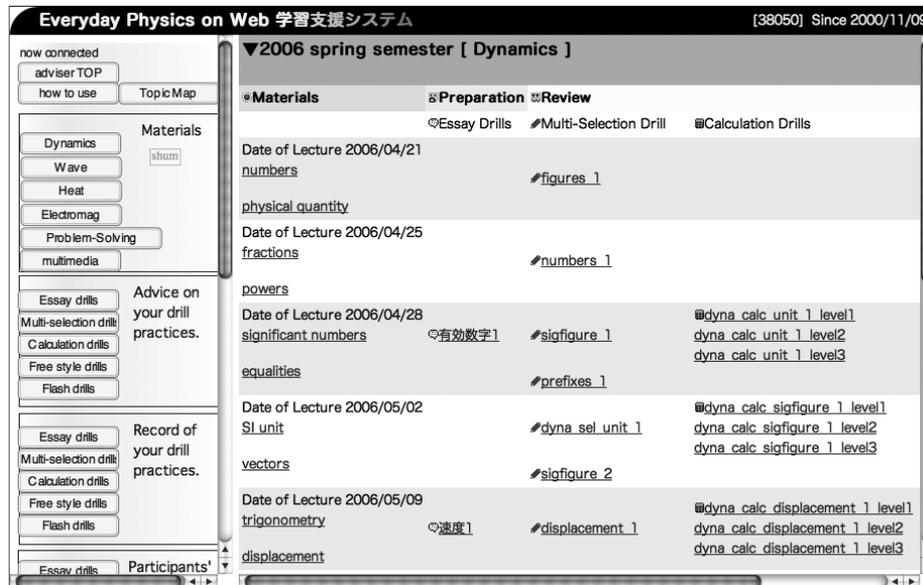


Fig. 1. An example of schedule type portal. Links for texts and drills were arrayed linearly along with the lecture schedule, from the top to the bottom of the portal page.

2 Towards a Subject-centric learning system

2.1 Change from “Course-Centric” to “Subject-Centric”

In the traditional course styled e-Learning systems, learning resources are arrayed sequentially along with the lecture time schedules. The way of arrangement defines the context of information that the course provides. Following the sequence of learning materials, learners acquire necessary base knowledge to proceed to higher steps in the course. However, less motivated students often lose enthusiasm in the middle of the course, feeling difficulties in understanding or in skills. Further, in order that the learners are able to explore the learning resources according to their own particular interest, another mechanism is required in the learning system.

On the other hand, students often study at home by using learning resources that exist on the web. The manner of online information retrieval was modeled as “berrypicking”, where the motivation of retrieval is often stimulated and is changed in the repetition of retrieval, browsing, and thinking [6]. This

2.2 Creating a Primitive Topic Maps Portal and Students' Response

In autumn 2006, a primitive Topic Maps portal as shown in fig. 2 was created and introduced in our e-Learning system, EPEL. Contents of the system ranges over the fields of “basic mathematics”, “dynamics”, “heat”, “wave”, and “electromagnetism”, in the form of texts, multiple selection drills, calculation drills, essay drills, and other free-style drills.

The web server used was Windows IIS, and the data base server was Windows SQL Server 2000. Connection of web application and the data base server was implemented using Macromedia ColdFusion MX7. The client application was developed using Macromedia Flash 8 Professional. The flash remoting technology was used for the communication between client Flash pages and the ColdFusion components implemented in the web server. In addition, Macromedia Flash Communication Server MX was used for several real time communication functions implemented in EPEL system.

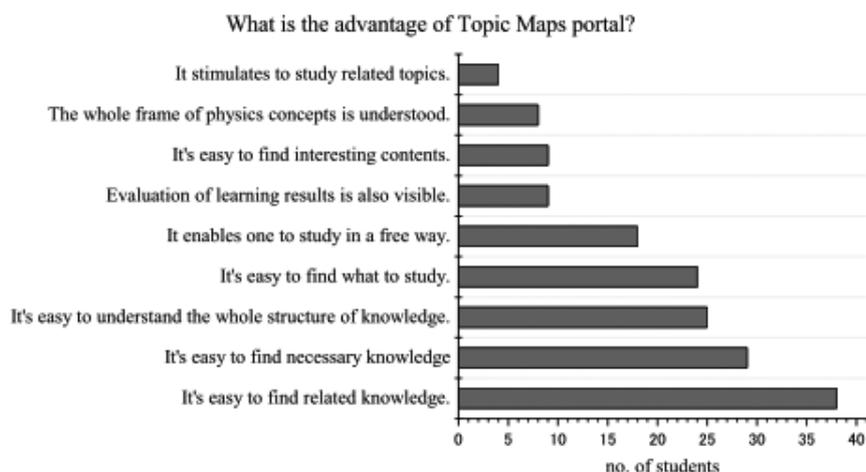


Fig. 3. Results of an inquiry for students, asking to choose useful pointsmerits of topic Topic map Maps portal, shown in fig. 2, from thea multiple selection list shown in the figure. Choice of more than one selection was allowed.

The primitive portal of EPEL consists of buttons with the labels of subject names, and association lines, which are emphasised with mouse rollover events, pointing from base subjects to applied subjects as also shown in fig. 2. The association role types are simply “base” and “application”, correspondingly. In most cases, the grain fineness of subjects is similar to the sub section of ordinary textbooks, such as “work”, “work and kinetic energy”. The way of association is based on ordinary traditional instruction method. Visually equivalent topic maps

were displayed for the above 5 types of contents. Maps were switched by a pull down menu, which was shown in the upper middle position of fig. 2.

To stimulate repetitive learning with an appropriate interval of time, 6 levels of advises were provided and visualized by the label colours of topic buttons. The levels of advise, from “Study again right now” to “You understood well, Try it after a while” were determined by the values of time-dependent weighted accumulation function of individual drill scores. These advise levels represent priority levels of reviewing.

Our e-Learning system, EPEL, was utilized in the introductory section of the curriculum of School of High-Technology for Human Welfare, which consists of Dept. of Perceptual Human Interface Design, Dept. of Information and Communication Engineering, Dept. of Materials Chemistry, Dept. of Biological Science and Technology, and Dept of Bio-Medical Engineering, in Tokai University. “Classical dynamics” courses (2 sessions every week) were opened in the spring semesters, and “heat and energy”, “oscillation and wave”, and “electromagnetism” courses (one session every week for each class) were opened in the autumn semesters. In 2006 and 2007, each class had approximately 30 students. Most of the students utilized e-Learning system during semesters.

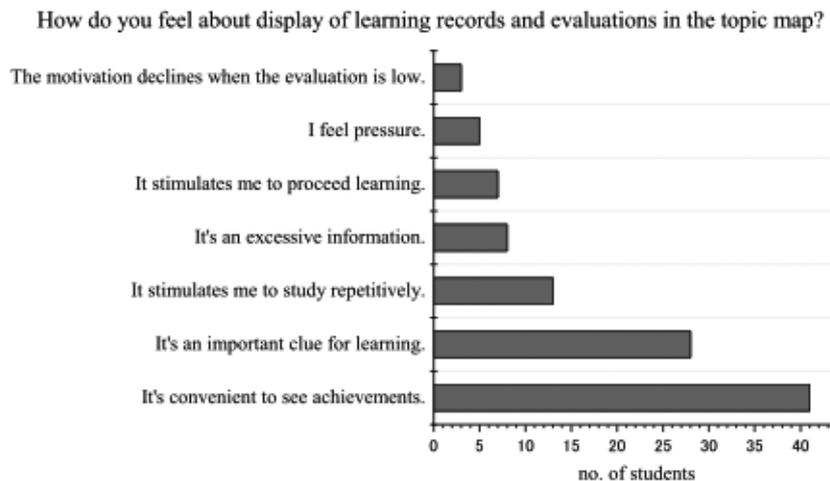


Fig. 4. Results of an inquiry about on impression of impression display of drill score evaluations displayed in the topic maps portal. Choice of more than one selection was allowed.

From autumn 2006 to spring 2007 semesters, the number of students who studied a wider range of fields in the e-Learning than in face-to-face class increased. Particularly, those who studied many times had a tendency to explore uniformly

on the map. This suggested that the topic map was effective for inducing self-study of non-lectured field. In addition, a positive feedback appeared in studying behavior from base knowledge to applied one [7].

Now we turn to the results of inquiry that was done for students in the middle of the autumn 2007 semester on their impression of the Topic Maps portal. Figure 3 shows the result for a question on the positive factors of the Topic Maps portal. Total number of students who answered was 71. Many students regarded the Topic Maps visualization had advantages in the recognition of relationships of knowledge. Not many students thought the maps were effective to grasp whole structure of physical knowledge. This might be due to the complexity of the visualization with too many items.

Figure 4 shows the result of the inquiry on displaying evaluation of drill scores on the Topic Maps buttons. Many of the students seemed to feel convenience in finding achievement and items to review. However, a number of students claimed complexity with excess information.

Figure 5 shows the results on the negative points of map representation. Many students claimed it was hard to see the order of study on the map. One of the student even claimed that the former scheduled type portal was better for preparation and review for the face-to-face class. Another student suggested that a fairly simple view was necessary at first to find the way to study easily, and, as the learning proceeded, the detailed presentation should appear gradually.

Balance between “Push” and “Pull”. Considering these results, one solution that should be made is the introduction of sequential navigation that corresponds to the traditional course structure. However, many of the students in our department are likely to make less effort to get through the course. Sufficient care should be taken to introduce simple navigation in the topic maps.

This problem concerns the balance of “push” and “pull” for the findability. If the direct navigation that works as “push” dominates in the system, learners might not become active in exploring the knowledge. This, in turn, might cause a decline of motivation. On contrary, if functions of “pushing” are removed, learners will not feel any stimuli to commence or keep learning activity.

Obviously, use of simple navigation will not solve this problem. The state of balance between “push” and “pull” will evaluate the learning system. “Push” and “pull” properties characterize every interactive function. Advise for repetitive study, as well as representation of drill scores, will act as “push” function in our system. Also, Topic Maps portal as a whole is characterized by its “pull” property. Considering the results of inquiry, our present system can be evaluated to be still insufficient in its “push” property.

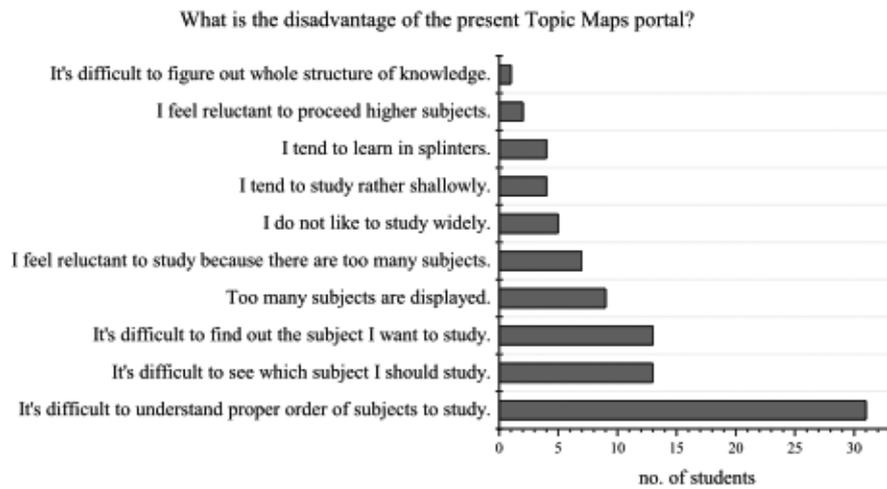


Fig. 5. Results of an inquiry about weaknegative points in the topic Topic maps portal. Choice of more than one selection was allowed.

3 Construction of Topic Maps for Subject Centric Extensible e-Learning

Our primitive EPEL system until 2007 did not fully make use of the potential of topic maps. Here we try to create a multi-layered topic maps to enable the system more extensible. We consider the following requirements.

1. Topic Maps ontology of physics knowledge can be created and extended independent upon the actual learning resources and learning record data.
2. It should be easy to add new types of contents, independent upon the physics knowledge structure.
3. Learning Records are divided into assessable and non-assessable type, and assessable type is divided into self-assessment and automatic-assessment type.

3.1 Topic Maps Ontology of EPEL

Table 1. Topic types and occurrence types of EPEL.

Topic Types (broader <-> narrower)		Occurrences
Physics Subject	Basic Mathematics	(empty)
	Common Concepts	
	Dynamics	
	Heat	
	Wave	
	Electromagnetics	
Learning Resource	Text	(SQL queries for learning resources)
	Essay Drill	
	Multiple-Choice Drill	
	Calculation Drill	
	Free Drill	
Learning Record	Text Learning Record	(SQL queries for users' learning records)
	Essay Drill Learning Record	
	Multiple-Choice Drill Learning Record	
	Calculation Drill Learning Record	
	Free Drill Learning Record	

Table 1 shows the taxonomy of topic types and occurrences for EPEL 2008 spring semester. Knowledge structures are considered in “Physics Subject” type. The structure of learning materials is considered in “Learning Resource” type. The recorded data of individual learning, such as the date and time of sessions, scores of drills, the values of time-dependent weighted accumulation function of drill scores, students’ comment and questions, etc., are considered in “Learning Record” type. Sub types of “Physics Subject” are divided into 6 individual field types, and each field has its own characteristic network of associations among its instance topics.

No occurrence is defined for “Physics Subject” type. One can concentrate on the concept of physics and construct ontology of physics subjects, even without thinking of any actual materials.

“Learning Resource” type has, at present, 5 types of materials. These types are distinguished by the format of learning materials. Thus, the ways of formatting are considered as topics in this e-Learning ontology. Since the ways of contents formatting are independent upon the subject of physics, “Physics Subject” type is independent upon “Learning Resource” type. Occurrences of “Learning Resource” and “Learning Record” are embodied in a variety of SQL queries.

“Learning Record” is the topic type for organizing individual records of activities in EPEL system, and is also based on the ways of assessments. In EPEL, text reading has no assessment. For essay drills, learners write texts, and then

compare with the example of answer. In this case, learners themselves assess their understanding, i.e., self-assessment. For other drills, learners' answers are automatically checked and analyzed by the evaluation function, i.e., interactive-assessment.

Table 2. Association types and association role types of EPEL.

category	Association Type	Association Role Type	
Physics Association	is_based_on	base	application
	Transfield_is_based_on		
	Advanced_is_based_on		
	Applied_is_based_on		
	is_analogous_to	(symmetric)	
Learning Resource Association	is_subject_of_Resource	previous	next
Learning Record Association	is_subject_of_Record	Subject	Text
			Essay Drill
			Multiple-Choice Drill
			Calculation Drill
			Free Drill
Learning Record Association	is_subject_of_Record	Subject	Text Learning Record
			Essay Drill Learning Record
			Multiple-Choice Drill Learning Record
			Calculation Drill Learning Record
			Free Drill Learning Record

Table 2 shows the association types and association role types for corresponding topic types. "Physics Association" category defines various types of associations among "Physics Subject" topic instances. "is_based_on" type and "Transfield_is_based_on" type define a hierarchical structure of physics knowledge system. "Advanced_is_based_on" association and "Applied_is_based_on" association connect the basic concepts with the advanced subjects or with the applied subjects. These associations are particularly useful for learners to explore the knowledge from the basic concept.

The navigation association “Preceding_Following” is used to simulate ordinary course learning system. It provides traditional order of learning materials to learn introductory physics. Many of the basic subjects within each field are connected sequentially by this association. As it was mentioned in section 2.2, many of those students who used EPEL claimed to make clear a standard order of subjects to learn.

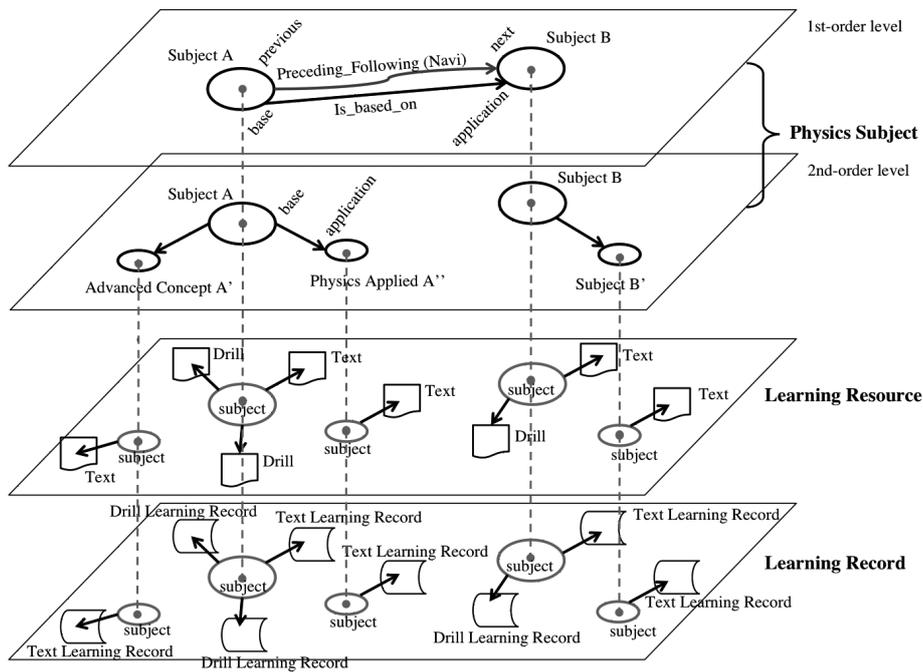


Fig. 6. Structure of topic maps. Physics subject types forms the knowledge structure of physics, without having occurrences to specific materials. Learning resource types and learning record types are related to the physics subject types, and have occurrences to real learning materials and records of learners’ activities and scores.

The topic map of instances of “Physics Subject” in the knowledge layer constructs the fundamental structure of EPEL. Instances of subtypes of “Learning Resource” type and “Learning Record” type are associated to the networked “Physics Subject” topics. Thus, the architecture is centered in physics subject in the knowledge layer.

3.2 Multi-Layered Structure of Topic Types

Figure 6 shows a schematic illustration of the structure of EPEL topic types, by a multi-layered modeling. The backbone of EPEL is the 1st-order level where the instances of basic subjects are connected by “is_based_on” associations. Also, many of basic subjects are aligned with “Preceding_Following” navigation associations. Advanced subjects and applied subjects are connected with the corresponding basic subjects in the 2nd-order level. The layers of physics subjects can be added by introducing another types of associations.

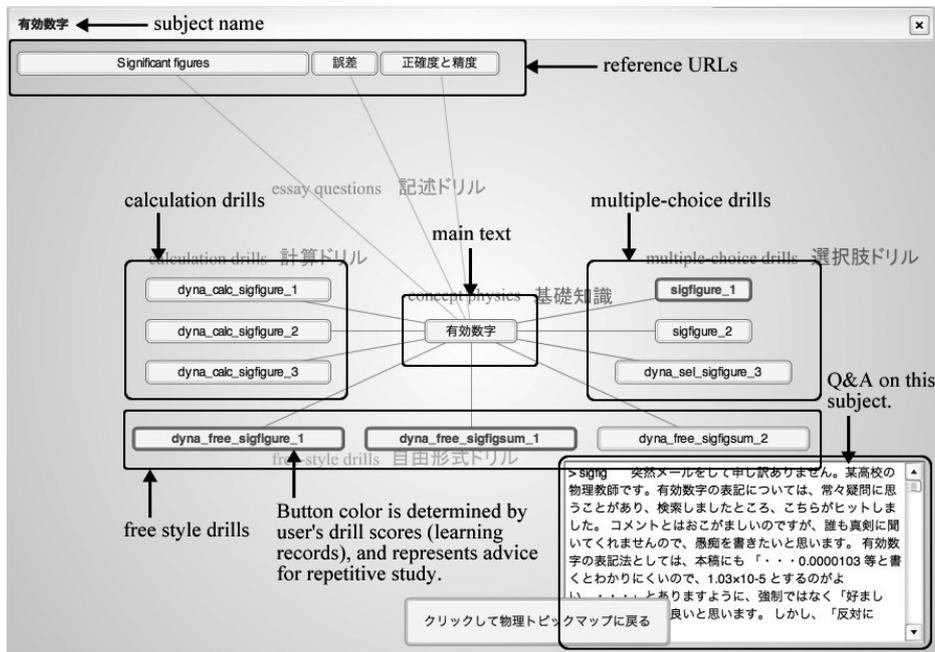


Fig. 7. An example of representation of 2nd-order level topic maps portal. A button for basic subject is located in the center, and those for drills are around. Buttons at the top are for related subjects.

Learning resource and learning record type instances are located on the corresponding layers, and associated with the physics subjects by “is_subject_of_Resource” and “is_subject_of_Record”, correspondingly. In the figure, resource type instances, as well as record type instances, are connected with subject images projected onto the corresponding layer. Learning resource layer and learning record layer have the occurrence links to the real materials and recorded learning data.

Figure 7 shows an example of real image of 2nd-order map layer, with the learning resource layer and learning record layer superimposed on it. Particularly, information of the learning record layer, i.e., the values of time-dependent weighted accumulation function, is represented by the button colors. Physics subjects at 2nd-order level themselves have no occurrences. Only the subject names appear as labels of buttons. The association between physics subjects are visualized by line. The main text instance button of basic subject is located at the center, and its text is downloadable by clicking the button. Text instances of related subjects are located at the top of window. Buttons for drill resources surround the main subject. Label colors of material buttons represent the data of learning records.

4 Concluding Remarks

This paper discussed the change of e-Learning system from traditional sequential course type to a primitive Topic Maps portal type. Students' responses to this change suggested that the Topic Maps portal was useful for recognizing relationships of knowledge, and that the balance between push and pull in the functions of e-Learning system should be improved. In the next step, we proposed a new topic map system that is physics subject centric as backbone ontology. One can concentrate on extending the physics subjects and the associations.

Learning introductory physics as efficiently as possible is really important for beginners who have a variety of majors. However, it is also important to bridge between basic physics knowledge to real natural phenomena and technological applications in students' self-study on web. The subject-centric nature of Topic maps technology will play an essential role for this purpose.

Acknowledgments. Work on this paper has been partially funded by Grant-in-Aid for Scientific Research (C) 19500760 from the Ministry of Education, Culture, Sports, Science and Technology, Japan, and by a grant from Nissan Science Foundation.

References

1. Matsuura, S.: <http://nkiso.u-tokai.ac.jp/EPEL/PhysElearning.html>

2. Mizuno, R: *Cognitive Psychology of Learning Effect*. Nakanishiya Shuppan Kyoto (2003)
3. Matsuura, S: Support of Repetitive Learning in an Elementary Physics e-Learning System. *Jpn. J. Educ. Technol.* 29(Suppl.), 193--196 (2005)
4. Dicheva, B., Dicheva, D.: Visual Browsing and Editing of Topic Map-Based Learning Repositories. In: Maicher, L., Sigel, A., Garshol, L.M. (eds.) *TMRA 2006*. LNCS, vol. 4438, pp. 44—55. Springer, Heidelberg (2007)
5. Lavik, S., Nordeng, T. W., Meløy, J. R., Hoel, T.: Remote Topic Maps in Learning. In: Maicher, L., Sigel, A., Garshol, L. M. (eds.) *TMRA 2006*. LNCS, vol. 4438, pp. 67—73. Springer, Heidelberg (2007)
6. Bates, M. J.: The Design of Browsing and Berrypicking Techniques for the Online Search Interface.
http://www.si.umich.edu/~rfrost/courses/SI110/readings/InfoFinding/Bates_on_Berrypicking.pdf (1989)
7. Matsuura, S.: Learning Trajectory on a Topic Map of Introductory Physics e-Learning. In: *Asian Topic Maps Summit 2007 Proceedings*, 161—164 (2007)

Topic map for Topic Maps case examples

Motomu Naito

Knowledge Synergy Inc., 3-747-4-203 Kusunokidai Tokorozawa,
Saitama 359-0037, Japan

motom@green.ocn.ne.jp
<http://www.knowledge-syergy.com>

Abstract. When developing topic maps and their applications, key challenges are how to pick up the main subjects in targeted domains and how to systematize those subjects. This paper introduces a topic map development about topic map case examples. It also introduces what kinds of subjects were extracted and how the identifiers of those subjects were given and how those subjects were classified in the first version. Then the difficulties which were emerged during the development are discussed. In order to promote sharing of the case examples and make good use of them, I provide some consideration and future works.

Keywords: Topic map development, subject, subject classification, subject systematization, Topic Maps case example

1 Introduction

Potentiality and practicability of Topic Maps attract many people increasingly. More and more Topic Maps case examples have been developing by many people. But it is difficult for us to search out the case examples which we really want to find. Many presentation documents have published from such as Topic Maps 2007, 2008, TMRA, AToMS conference web site. Mostly those web sites only enumerate the abstracts of presentation and have links to the presentation documents. If those presentations can be navigated and accessed according to specific subjects and viewpoints, convenience, availability and usefulness of those web sites will increase significantly.

Many people often ask what kinds of case examples exist and where those case examples can be found. In order to answer those questions, to share the case examples in Topic Maps community and with new comers, and to find expected case examples easily, I have been developing a topic map for Topic Maps case examples. At the start of the development, 67 presentations at Topic Maps 2007, TMRA 2007 and AToMS 2007 were targeted. According to Steven R. Newcomb [5], Topic Maps activity started to try to make master index for many documents. Similarly this try can be said as the effort to make master index for Topic Maps related presentations, activities, products, etc. throughout cyberspace.

In this paper, I address the challenges how to classify Topic Maps case examples and how to make easy to find target case examples. In the domain, I consider what main subjects are and how to organize those subjects. Concerning the topic map about Topic Maps case examples, the remainder of this paper is the following. In section 2, analogical topic maps are given. In section 3, developing process and the result until now are described. In section 4, problems are studied. Finally conclusion and future work are showed in section 5.

2 Related topic maps

Topic map for conference proceeding was provided at XML 2001 conference in Orlando, USA. And it got into the news of those days. That topic map had topic types such as author, presentation title, country, keywords, etc. and it could be navigated from the viewpoint of those topic types. The topic map did not include information of multiple conferences but only included information of that conference.

TOPICMAPS.COMMUNITY's website [9] is the website for Topic Maps related information. And itself developed based on topic map. It is possible to navigate and access the presentation documents used in conferences such as Topic Maps 2007 and 2008 in Norway. It also has links to other Topic Maps conferences such as TMRA and AToMS. It doesn't have the master index across a number of those conferences.

Fuzzzy.com's website [10] has much information about Topic Maps Portal and Topic Maps Online Application. It has short descriptions of those portals and applications and has links to those sites. It has many tags from some viewpoints but those are not categorize and organized.

All of the above cases don't have sites crossing index. In this paper I will describe one trial to make master index, in other words topic map, across to multiple website.

3 Topic Maps case examples topic map and its web application

In this section, making process, the topic map and its application are described.

3.1 Making process

The first version of the topic map for Topic Maps case examples were created by the following processes.

1. Data collection and analysis
2. Ontology making
3. Topic map making
4. Application development

Data collection and analysis In the first stage, targeted data was 67 presentations at three conferences, Topic Maps 2007, TMRA 2007 and AToMS 2007. Those data was collected on EXCEL with CSV format by hand. The collected items (subjects) can be roughly divided into two groups, the fact data group and the data group picked out from presentation document based on my understanding. The items in fact group are Events, Sessions, Presentations, Persons, Countries, Organizations, and so on. And the items in picked out group are Activities, Products, Purposes, Industrial domains, Target information/knowledge, Providing services, Activity entities, Users, and so on. These items would be strong candidates of the subject for the topic map. The reason those items were chosen was those items seemed to become good view point to navigate the topic map and to find wanted case examples.

Ontology making Ontology was made according to the collected items (subjects) and relations between them. Fig. 1 shows the ontology diagram. In Fig. 1, topic types are represented by squares and association types are represented by lines.

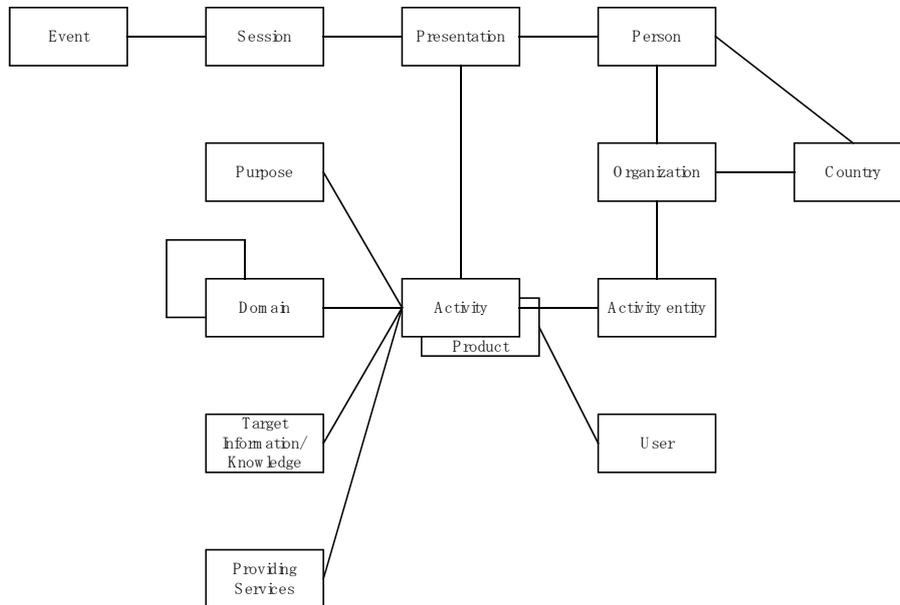


Fig. 1. Ontology diagram of Topic Maps case examples topic map

In this trial, it was differentiated the Presentation topic type and the Content topic type of presentation (it was represented Activity topic type or Products topic type.) Each instance of the Presentation topic type identifies the each presentation. And it has close relations between topic types such as Event, Session, Person, Country and Organization. Each instance of the Content topic type identifies the each Activity or Product mentioned by the presentation and it has close relations between topic types such as Purpose, Industrial domain, Target information/knowledge, Service providing, Activity entity, Users.

Topic map making Using the collected data as input and based on above ontology, the Topic Maps case examples topic map was generated. Specifically, the topic map was generated using the DB2TM module included in OKS (Ontopia Knowledge Suite)TM [6]. The details of the topic map are described in section 3.2.

Application development In order to display and navigate the topic map, a web application was developed. The web application was developed according to the topic map and using Navigator Framework function of OKS. The details of the web application are described in section 3.3.

3.2 Topic Maps case examples topic map

In order to use DB2TM module, the ontology definition file and the XML configuration file were made. Topic types, Association types, Association Role types and Occurrence types were defined in the ontology definition file with LTM [7] format. The mapping rules from collected data with CSV format into the ontology definition file were described in the XML configuration file. And then according to the ontology file and the XML configuration file, Topic Maps case examples topic map was generated by batch process of DB2TM module.

As topic type there are Event, Session, Presentation, Person, Country, Organization, Activity, Product, Purpose, Industrial domain, Targeted information/knowledge, Providing services, Activity entity and User in the topic map. Gathered information is 67 presentations from Topic Maps 2007, TMRA 2007 and AToMS 2007. And structure of the topic map was corresponded the Ontology which was showed in Fig 1. At the moment, version 1.0, the numbers of Topic Maps components were showed in Table 1 and Table 2.

Table 1. The number of types

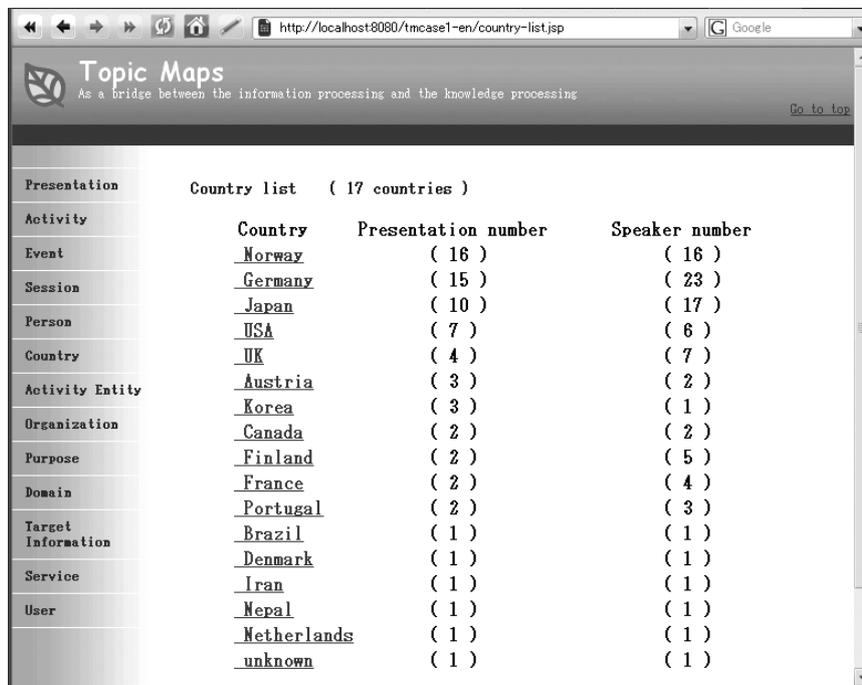
Type	The number of types
Topic	17
Association	15
Association Role	30
Occurrence	1

Table 2. The number of Instances

Instance	The number of instances
Topic	682
Association	1094
Occurrence	67
Total	1843

3.3 Topic Maps case examples topic map application

The application was developed using OKS Navigator Framework. The Navigator Framework is based on the Java 2 Platform, Enterprise Edition (J2EE), using the Java Servlets and Java Server Pages (JSP) technologies. It is said that OKS makes possible to develop Topic Maps based web applications rapidly and easily. It consists of a set of JSP tag libraries, and a Java API. Applications developed with it can be deployed into any J2EE container. In Topic Maps web applications, we can navigate related topics according to associations in a subject centric way.



Country	Presentation number	Speaker number
<u>Norway</u>	(16)	(16)
<u>Germany</u>	(15)	(23)
<u>Japan</u>	(10)	(17)
<u>USA</u>	(7)	(6)
<u>UK</u>	(4)	(7)
<u>Austria</u>	(3)	(2)
<u>Korea</u>	(3)	(1)
<u>Canada</u>	(2)	(2)
<u>Finland</u>	(2)	(5)
<u>France</u>	(2)	(4)
<u>Portugal</u>	(2)	(3)
<u>Brazil</u>	(1)	(1)
<u>Denmark</u>	(1)	(1)
<u>Iran</u>	(1)	(1)
<u>Nepal</u>	(1)	(1)
<u>Metherlands</u>	(1)	(1)
<u>unknown</u>	(1)	(1)

Fig. 2. Country's web page

The application consists of about 20 JSP programs at the moment. It makes possible to start to navigate inside the topic map from each topic type described in section 3.2. It can display the instance list of each topic type and the details, (namely associations and occurrences) of each instance topic. It also has the functions such as Sort, Count, Full text search, Graphical representation and so on. Fig. 2 shows Country's web page generated by the application. Fig. 3 shows graphical representation of the topic map.

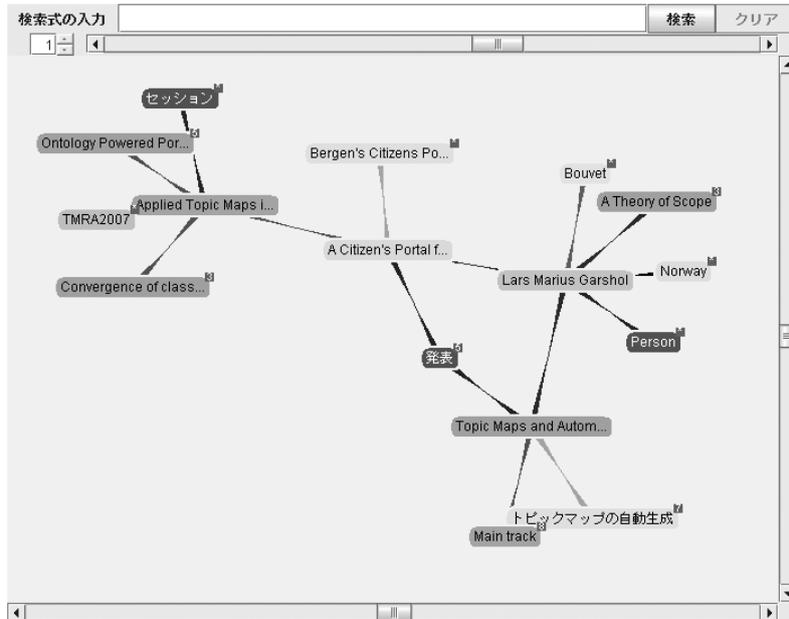


Fig. 3. Graphical representation of the topic map

The application brings many interesting results. Table 3, Table 4, Table 5 and Table 6 show Country basis ranking, Person basis ranking in other words the most frequent speaker ranking, Organization basis ranking and Industrial basis ranking for each. Norway wins the first prize. The most frequent speakers are Lars Marius Garshol, Markus Ueberall, Michihiko Setogawa and Sam Gyum Oh. The most frequent organization is Bouvet. The top industrial domain is Information and Communications. Those are results of tolog [8] query with the functions such as Sort and Count and be picked up from the web application.

Table 3. Country basis ranking

Ranking	Country	The number of presentation
1	Norway	16
2	Germany	15
3	Japan	10
4	USA	7

Table 4. Person basis ranking

Ranking	Person	The number of presentation
1	Lars Marius Garshol	3
1	Markus Ueberall	3
1	Michihiko Setogawa	3
1	Sam Gyun Oh	3

Table 5. Organization basis ranking

Ranking	Organization	The number of presentation
1	Bouvet	8
2	Hitachi System and Services	3
2	J.-W.-Goethe University	3
2	National Institute of Informatics	3
2	Networked Planet	3
2	Ontopedia	3
2	Sungkyunkwan University	3
2	University Leipzig	3

Table 6. Industrial domain basis ranking

Ranking	Industrial domain	The number of presentation
1	Information and communications	38
2	Education-Learning support	15
3	Government	7

4 Issues and discussion

Some issues became clear through this experience. Those are regarded as general problems in topic map creation.

4.1 Coding scheme of Subject Identifier

The first issue is what kind of coding scheme is suitable for Subject Identifier, especially for fragment of IRI (Internationalized Resource Identifier). It is easy to assign serial number within the limits of collected data. But generally it is not a good way because it lacks the generality, the scalability and the reusability, and it is not intuitive and not friendly for human. If there is already authorized code system such as country code, it is appropriate to use it. The issue is in the case of we can not find those code system, for example presentation identifier, person identifier, etc. I am using conference name + serial number for presentation identifier (example: TMRA2007-1, TMRA2007-2...) and family name + first name for speaker identifier (example: MaicherLutz) at the moment, but these code systems don't seem the best. Those identifiers include some problems such as synonym and homonym problem. I have to seek after a better system from now on.

4.2 Classification scheme

In this work the most difficult part is to build a classification system. If I can categorize some subject according to human's conceptual system, it seems to become easy to navigate intuitively along the classification system. If I know the existence of a suitable classification system, I can use it. If I don't know the existence of that, I have to devise a classification system by myself. I could find a suitable classification system for Industrial domain for this work. I used Japan Standardized Industrial Classification. I needed only to map the industrial domain of presentations to the classification. Therefore it was relatively easy work. Japan Standardized Industrial Classification is four layered classification. Those layers are L category, M category, S category, T category. The L category is the following:

- A: Agriculture
- B: Forestry
- C: Fisheries
- D: Mining
- E: Construction
- F: Manufacturing
- G: Electricity-Gas-Heat supply and Water
- H: Information and Communications
- I: Transport
- J: Wholesale and Retail trade

K: Finance and Insurance
L: Real Estate
M: Eating and Drinking places- Accommodations
N: Medical - Health Care and Welfare
O: Education-Learning support
P: Compound Services
Q: Services- N.E.C.
R: Government- N.E.C.
S: Industries unable to classify

I could not find a suitable classification system for Activity, Purpose, Targeted information/knowledge, Providing service, etc. Therefore the classifications of those subjects were very difficult. For the work I assigned appropriate words for the subject of the presentations, and then made effort to classify those words. I think those are similar process to KJ Method. I'd like to continue the effort to search for the method of classification as well as to develop the good method to build classification system.

4.3 Appropriate metadata for posting

I picked out suitable words from presentation documents for Purpose, Targeted information/knowledge, Service providing, User, etc. from my point of view. And still I'm taking great pains over building of those classification systems. Like as Industrial domain, if there are good classification systems authors can select and attach suitable category to their activities as metadata. They can publish the activities with the metadata. In result more appropriate classifications become possible. Therefore I think publication with appropriate metadata is very meaningful. I think we need to construct and share common vocabularies for those metadata.

5 Conclusion and Future Work

As the first step of the developing, it became possible to navigate 67 presentations from the three conferences. It also became possible to access those documents easily and I can use the topic map for my Topic Maps activity usefully. I can reply the questions about Topic Maps case examples in my Topic Maps popularization activity.

As future work I am planning the following:

- Review and improve the ontology
- Add for more viewpoints
- Review and improve Identifier coding scheme
- Review and improve Classification system

I think those works lead to improvement the topic map itself. After the topic map and its application make the pass mark, I'm planning to open the topic map through website as well as add more presentation from other conferences, an individual case examples and others one after another. Moreover I wish it becomes possible to discuss classification system and cooperate and merge with other topic maps in the open environment.

References

1. Steve Pepper: Topic Maps for the Three Kingdoms : The Many Applications of Topic Maps, ATOMS Asian Topic Maps Summit, 2nd June, 2006. pp 37-64
2. Steve Pepper: The TAO of Topic Maps : Finding the Way in the Age of Infoglut, <http://www.ontopia.net/topicmaps/materials/tao.html>.
3. Steve Pepper: As We REALLY May Think : Memex, Topic Maps, and subject-centric computing, <http://www.ontopedia.net/pepper/slides/ATOMS2007.ppt>.
4. Steve Pepper, Sylvia Schwab: Curing the Web's Identity Crisis : Subject Indicators for RDF, <http://www.ontopia.net/topicmaps/materials/identitycrisis.html>.
5. Jack Park, Sam Hunting: XML Topic Maps : Creating and Using Topic Maps for the Web, ISBN0-201-74960-2.
6. Ontopia: Software Products : The Ontopia Knowledge Suite, <http://www.ontopia.net/solutions/products.html>.
7. Ontopia: The Linear Topic Map Notation : Definition and introduction, version 1.3, <http://www.ontopia.net/download/ltn.html>.
8. Ontopia: tolog : Language tutorial, <http://www.ontopia.net/omnigator/docs/query/tutorial.html>
9. TOPICMAPS.COMMUNITY : <http://www.topicmaps.com/tmc/>
10. Fuzzy.com : Topic Maps Portal or Online Application <http://www.fuzzy.com/tag/?id=2238>
11. Riichiro Mizoguchi: Science of Intelligence: Ontology Engineering, Ohmsha, ISBN4-274-20017-5.

12. Motomu Naito, Hiroyuki Kato, Takashi Kiriyama, Yushi Komachi, Mi Setogawa, Keiji Nakabayashi, Mitsuo Yoshida: An Introduction to Topic Maps, Tokyo Denki University Press, ISBN4-501-54210-1.
13. Lars Marius Garshol: Metadata? Thesauri? Taxonomies? Topic Maps: Making sense of it all!
<http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html>.
14. Frederic Andres and Motomu Naito: Dynamic Topic Mapping Using Latent Semantic Indexing" in IEEE International Conference on Information Technology and Applications Sydney 4th-7th July, 2005. pp 220-225
15. Motomu Naito and Frederic Andres: Application Framework Based on Topic Maps, First International Workshop on Topic Maps Research and Applications, TMRA 2005 Leipzig, Germany, October, 2005. pp 42-52
16. Sachit Rajbhandari, Frederic Andres, Motomu Naito, Vilas Wuwongse : Semantic-Augmented Support in Spatial-Temporal Multimedia Blog Management, Second International Conference on Topic Maps Research and Applications, TMRA 2006 Leipzig, Germany, October, 2006. pp 215-226

Topic Maps and Social Software

Connecting Topincs

Using transclusion to connect proxy spaces

Robert Cerny

Anton-Kubernat-Straße 15, A-2512 Oeynhausen, Austria

robert@cerny-online.com
<http://www.cerny-online.com>

Abstract. Topincs is a software system for agile and distributed knowledge management on top of the common web infrastructure and the Topic Maps Data Model. It segments knowledge into stores and offers users a way to establish a temporary connection between stores through transclusion and merging. This allows to easily copy topics and statements. Through this mechanism, later acts of integration are simplified, due to matching item identifiers. Furthermore, transclusion can be used to create permanent connections between stores.

1 Introduction

A Topincs store [5, 6, 10] is a knowledge repository for an individual or a group to make statements about subjects. It uses the Topic Maps Data Model [7] for representing knowledge. Throughout this paper we will illustrate crucial aspects by the example of a small fictional software company, called Modestsoft. It employs around 10 people and has various departments, including development, testing, and sales & services.

Modestsoft uses several shared Topincs stores to manage its organizational knowledge. Those stores deal with Modestsoft's *staff*, *products*, *issues*, *clients*, and their *configurations*, to name only a few. Additionally, every employee has his own store. While the shared stores have a clear boundary around their domain, the personal stores may contain topics and statements of a wider variety and are considered a repository for note taking, and as such they function as a memory extender [4]. Figure 1 illustrates Modestsoft's setup and the Topincs stores they use to manage their spontaneous knowledge recording demands.

A Topincs store qualifies as a knowledge node in a distributed knowledge management system as laid forth by Bonifacio et al. in [3]. It offers *semantic autonomy* for a person or group to be able to encode knowledge in a topic map. Topincs is a server application exposing a web interface. The current implementation uses PHP to process requests, runs on top of Apache, and uses MySQL for persistence. The two existing clients are browser-based JavaScript applications that address different user needs, but operate on the same repository:

- Topincs Editor: A topic map editor which offers maximum expressivity and requires in-depth knowledge of the Topic Maps paradigm. It is necessary to state prototypical statements that pave the way for ontological reflection.
- Topincs Wiki: A semantic wiki for quick editing with limited expressivity for people with intermediate computer skills and little to no knowledge of Topic Maps.

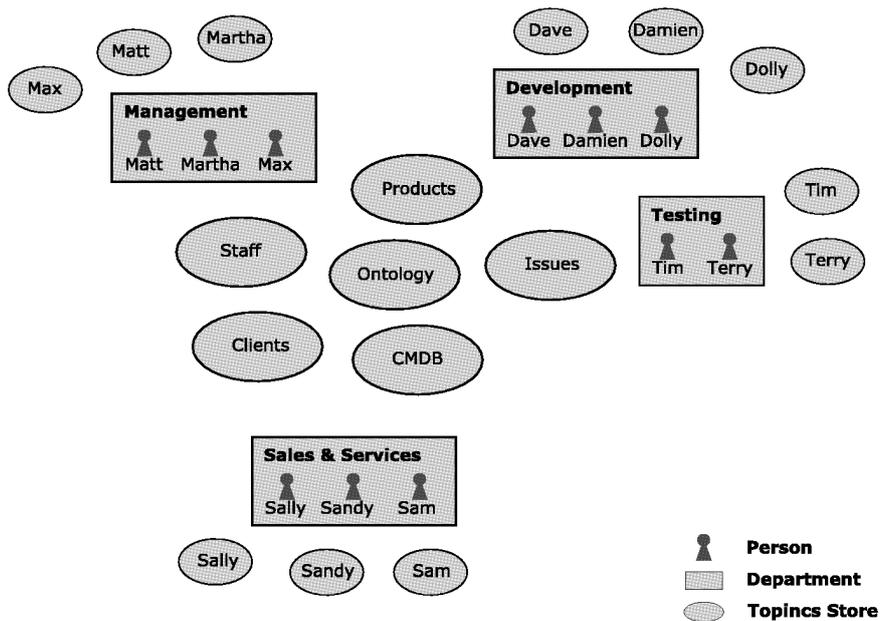


Fig.1. A schematic overview of the knowledge landscape of Modestsoft, a fictional software company with agile and distributed knowledge management.

Both clients offer specialized input elements to ease the creation of statements and make the topic map editing process similar to filling out a form. Yet, the Editor has a strong affiliation to the Topic Maps terminology which makes it unusable for most people. The Wiki avoids such terminology and allows only

statements of a known form and is therefore usable by a wider audience. It extracts a topic map fragment for the subject of the Wiki page from the store by an algorithm similar to the one defined by Ahmed [1] and renders this fragment in the browser for viewing and editing. New statements about a subject can be issued one by one. The next step in this evolution of restricting expressivity leads to Topincs Forms, which allows the grouping of statements and simple validation for more control over what should and can be said.

Topincs currently uses *ontological reflection* to suggest statement types to the user and to determine the datatype of an occurrence [6]. In a later version it will be instrumented to restrict the players of roles to topics of certain types. This mechanism makes Topincs a tool for agile knowledge management where encoding demands can be satisfied within minutes as they occur, without the need for detailed planning and programming. A user simply has to open the Editor, which has no restrictions and allows him to issue the new prototypical statement. Subsequently, due to *ontological reflection*, the statement type will be suggested in the Wiki. One Topincs installation can host any number of stores which are identified by an URL relative to the domain. Modestsoft uses `/staff`, `/products`, `/clients`, and `/cldb` for its shared stores.

This paper discusses a solution to easily exchange topics and statements between stores with the help of transclusion. This exchange allows *coordination* between knowledge nodes, which, besides *semantic autonomy*, is demanded as the second criterion for a distributed knowledge management system by Bonifacio et al. [3]. Without this solution Modestsoft had to create the topic type *Person* manually in every shared and group store and assign the subject identifier every time. With the work presented in this paper it is possible to create all typing topics in one designated ontology store and copy it into other stores. Once a topic is copied, it is easy to update it and fetch new statements.

The isolation of the proxy space is necessary to enable a search with acceptable response time when users need to refer to a subject they have in mind. It has the additional benefit that the user is forced to create a proxy for the subject which is under his control. By creating an item, Topincs automatically assigns an item identifier, an URL, which can be used to retrieve and manipulate the item by the HTTP methods POST, GET, PUT, and DELETE [5].

By using a Published Subject Identifier (PSI), the fact that two topics represent the same subject can be established manually. This paper does in no way intend to discourage the use of PSIs, on the contrary, it presents an alternative that relieves the users of the manual assignment and still lets them benefit from the mechanism that the Topic Maps Data Model provides for establishing sameness of subject for two proxies, by the means of item identifiers [7].

This paper introduces three forms of connecting Topincs stores and illustrates them with use cases. Furthermore it discusses the problems that arise with this approach and gives an outlook on future work.

2 Temporary Transclusion

Ted Nelson coined the term *transclusion* in his book *Literary Machines* [9]. Transclusion means the composition or augmentation of a document by including other documents or parts thereof by reference. It can be applied to documents that are written in natural language or a markup language. In both cases, it is important that the transcluded fragments can be interpreted, even though they are taken out of their original context. In our Topic Maps use case, it is best described by a transient merge map statement which triggers semantic integration of topic maps with some small differentiations depending on the type of transclusion.

The application of temporary transclusion in Topincs is the following: A user browses a foreign Topincs Wiki and discovers a page about a subject, for which a topic in his Topincs store exists. He decides to temporarily include the foreign page into his own page in order to see both merged. The result of this process is his page about the subject augmented with the foreign statements, which can be recognized visually. This temporary transclusion has no effect on the proxy space, yet it forms the base for two stronger connection types, namely *permanent transclusion* and *item copying*.

3 Permanent Transclusion

This form of connection allows the user of a store to permanently connect one topic to another topic representing the same subject in a different store. Whenever someone browses the page with the permanent transclusion directive, the included topic map will be retrieved and merged in the web browser at access time. The user will be able to recognize the foreign statements. Clicking on the player of a foreign association will lead the user into the respective proxy space given that there is no proxy in the local store. Updates in the foreign page will be reflected after a page reload.

The number of permanent transclusion directives on a page is not restricted. It is also possible to transclude arbitrary structured information in a page by incorporating a transformation step in the process.

4 Item Copying

From any form of transclusion, a user with edit rights on the Topincs Store can decide to save foreign statements into his proxy space. Since this process is very similar to manually creating statements, the existing infrastructure for editing and saving statements in the Wiki edit panel is used. Yet, the user has also the possibility to merge foreign topics with local ones. With a statement, all topics that are referred to within the statement, e.g. in the properties *type* or *scope*, are copied as well.

The screenshot shows the edit panel for the item 'Dolly' (Person). At the top right, there are links for 'view', 'edit', and 'include'. Below the title, there are buttons for '+ New statement', 'Selection', 'Preview', and 'Save'. The main content area contains several property fields:

- Date of birth:** 06/23/1978 (with a calendar icon), marked as 'New' (white N on blue background) and checked.
- Topic name:** Dolly (with a search icon), marked as 'New' (white N on blue background) and unchecked.
- Instance of:** Person (with a search icon), marked as 'New' (white N on blue background) and unchecked.
- Reports to:** Dave (with a search icon), marked as 'New' (white N on blue background) and checked.
- Has skill:** A list of skills: Java (marked as 'New' and unchecked), Ruby (marked as 'New' and unchecked), Unix (marked as 'New' and checked), and Windows (marked as 'New' and checked).
- Employed by:** Modestsoft, Inc. (with a search icon), marked as 'New' (white N on blue background) and checked.
- Has assignment:** Refactor all literal strings into constants (with a search icon), marked as 'New' (white N on blue background) and unchecked.

Fig.2. Dave transfers statements about Dolly from the staff store into the issues store. New statements are marked by a white N (for *New*) on blue background. All foreign statements from the staff repository have a grey background. All statements local to the issues store have a white background. Dave has selected uninteresting statements for deletion and already assigned Dolly her first issue.

By copying statements and topics, the user gains control over the items. He can then form his own statements about the new subjects and even correct copied statements. In the copying process the item identifiers are transferred, so that on subsequent views of the page it is possible to update statements. On a page with permanent transclusion in place, this will happen automatically due to matching

item identifiers. Without permanent transclusion, the user has to manually initiate the update.

In particular, the copying of topics of the ontology is beneficial for later integration of information between the two stores, since the need for manual merging will be significantly less and any act of transclusion will present a more homogeneous page. Within a Topincs installation, or even over several installations, it is recommended to have a designated store for the ontology, where users can fetch their topic types, occurrence types, name types, association types, and role types. With this store in place, users do not have to worry about later integration, given that the usage of the types is correct. The only practical method to enforce the correct usage of a proxy must be modeled after the mechanism used for learning of natural language. If the community observes the incorrect usage of a type, it must be corrected, like we correct little children when they are using a word incorrectly.

At a later stage, with Topincs Forms, a stronger enforcement of the correct usage can be in place. Topincs Forms is a framework for developing forms that generate sets of statements that are fed to a Topincs store via the web interface. This addition to Topincs is only in a very early planning phase. Its purpose, though, is clear. While the Topincs Editor offers complete freedom regarding which statements a user can issue, and the Wiki restricts users to statements of a certain form, Topincs Forms will put even stronger restrictions in place on which statements have to be said, and how these statements should relate to each other. This convergence brings us closer to achieving highly structured data which are not only useful for human consumption, but can be consumed by programs as well.

5 Use cases

5.1 Temporary transclusion and item copying

Our lead developer, Dave, wants to assign an issue to his newest team member Dolly. On the person page of the staff store, he can quickly transfer the statement about Dolly's existence into the issues store where assignments are recorded. On the include panel of Dolly's page in the issues store he can request the transclusion of the topic map about Dolly in the staff store. Dave switches to the edit panel and deletes those statements about Dolly that are not interesting in the context of software issues, e.g. Dolly's date of birth, and keeps the rest, e.g. which programming languages Dolly knows. He manually creates the issue assignment statement. Figure 2 shows what Dave has done so far. He finishes the

transfer by saving the page. From now on there is a topic representing Dolly in the issues store and this proxy holds references to other proxies that represent the same subject. The transclusion of Dolly's staff store topic map can be requested by Dave on the include panel of her Wiki article at any time to see whether statements were corrected or new statements were added.

5.2 Permanent transclusion and agile extension

Dolly is an experienced programmer. Developers in her old company used the issue tracking system to record the time they spent on issues. She knows the planning benefits of such information and wants to do the same thing in Topincs. She is not quite sure yet how to do it, so she decides to do a test run in her personal store. She visits her article page in the issues store and temporarily includes her page into the one in her personal store. She makes this transclusion permanent, so that she can easily copy new issues into her store in order to make effort tracking statements about issues assigned to her. This permanent transclusion allows her to learn about new assignments in her own personal store, although they are made by Dave in the issues store. Figure 3 shows her page in her personal store.

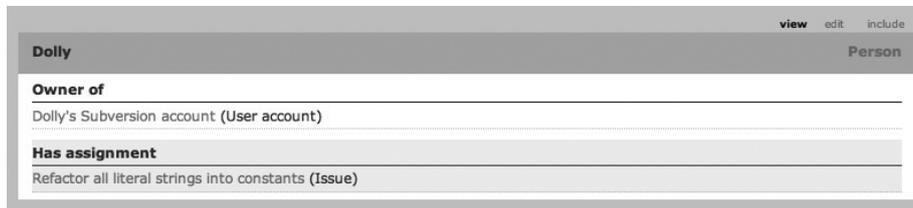


Fig. 3. Dolly's article in her own store. She has already noted that she owns a Subversion account. The assignments are retrieved from the issues store by permanent transclusion. Even the association type *Has assignment* is indicated as foreign with a grey background.

She analyses which topic and statement types are necessary for a minimal effort tracking system and comes up with the following: one topic type *Work session*, three occurrence types *Start time*, *End time* and *Summary*, and two association types *Participation* and *Realization*. The first association type is to relate a work session to the worker and the latter is to relate a work session to the issue that is processed. The creation of these types takes her approximately fifteen minutes including the issuing of the prototypical, pre-reflection statements in the Editor. She makes it a habit to start work by creating a work session with the statements *Start time*, *Participation*, and *Realization* and to finish it by stating the *Summary* and *End time*. Later she tells her boss Dave about her extension and he decides

that all developers should track their efforts. In order to keep concerns separated, he creates a new Topincs store `/efforts` and asks Dolly to copy her Work sessions into the Store. Due to ontological reflection, other users have immediate access to the statement types for work sessions in the Wiki.

6 Problems

By using transclusion we have gained an easy, quite simple way to copy items from one store to another. This handy mechanism, however, enlarges one problem which already exists within one repository: how to enforce the correct interpretation of a topic. Our lead developer Dave creates an occurrence type *Resolving date*. He intends it to be used by his developers to state that an issue was resolved on a certain date. He has to communicate his intent by creating a description. Within Modestsoft, the existence of such a type and its meaning can be announced in meetings. All humans, whether they consume the occurrence type on the Wiki page of an issue or whether they write programs using this type, participate in a social contract on when and how to use this topic. If the proxy leaves the boundaries of this social contract, in our case the office of Modestsoft, the likelihood of misinterpretation increases.

While this is a serious problem, it is one that affects any form of user interface. When completing a form, a user encounters as many questions as the form has fields. He interprets a question and, based on this interpretation, completes the field. Organizations use trainings to ensure the correct interpretation of the questions that their software asks users. It is hard to imagine a process with less control that achieves the same result. The problem magnifies if one considers global knowledge interchange. The domain has no boundaries — *people can talk about anything* — and users do not come from one cultural background and thus do not share an implicit context which supports them in their interpretation.

Besides unintentional misinterpretation of topics, there is the possibility of intentional misinterpretation. Something similar can frequently be observed in long running projects using relational databases: The meaning of a table column is altered or enriched. Usually, this act is justified by the high organizational costs of introducing new columns. Since the costs of introducing new statement types in a Topic Maps system are much lower and distinct proxies can always be merged, such intentional misinterpretation should be avoided at any price.

With the presented mechanism, it is very easy for a Topincs user to declare subject sameness for two proxies. Unfortunately, a wrong decision is difficult, nearly impossible to revert in an early merging approach that Topincs uses since incoming statements always use the current subject identity status to decide

which topic a statement is attached to. Even if erroneous subject sameness is detected and corrected, the statements about the *other subject* remain at the incorrect topic. A system with late merging, where all topics are considered distinct when storing a statement, would not have this issue.¹

The current transclusion mechanism is implemented by requesting a topic map in Json Topic Maps format [5] using the JavaScript object `XMLHttpRequest` in the Wiki article page. Due to security restrictions of the web browser this only succeeds within the same domain. Since this issue affects many Ajax applications, there are several work-arounds available and a standard approach for Cross-Domain Ajax requests is work in progress.²

7 Conclusion and Future Work

Due to the methods presented in this paper, Topincs stores are no longer isolated islands of knowledge, but offer mechanisms for the exchange of topics and statements. Topincs users can benefit from the integration mechanism of the Topic Maps technology without using PSIs, but rather by a simple item copy procedure. By copying topic and statement types the manual work necessary to create a distributed knowledge landscape and to integrate information is significantly reduced.

The declaration of subject sameness of a foreign topic and local topic in the store where the transclusion takes place should be supported by an automated detection mechanism. Maicher's SIM Approach [8] looks very promising in this respect.

Since established organizations have most of their information stored in arbitrary formats we would like to adopt Barta's idea of resource wrappers producing virtual maps [2]. In our case, such virtual maps would ideally contain item identifiers, so that statements can be recognized. A partial satisfaction of the Topincs Web Service Interface [5] regarding reading requests on item identifiers would allow even tighter integration of such virtual maps into Topincs.

With the exchange mechanism in place, the manual labor to manage personal or organizational knowledge in multiple Topincs stores is reduced since topics and statements can simply be copied from one repository to another, maintaining their identity, and easing later integration tasks. Nonetheless, Topincs should be made more user friendly, by further lowering the requirements to encode knowledge in a Topincs store. The most promising approach to this issue is the introduction of forms. Modestsoft chief tester Tim and his team always have to

¹ Personal conversation with L. Maicher and B. Bock in Leipzig in February 2008.

² <http://www.w3.org/TR/access-control/>

issue the same group of statements when they create an issue: *Reported by*, *Reporting date*, *Affects component*, *Discovered in version*, and *Description*. When a developer fixes a bug he has to state: *Resolved by*, *Resolving date*, *Resolved in version*, and *Comment*. In the Wiki, these statements have to be issued one by one, which is cumbersome in such a frequent task. Two new features will ease this activity: 1) statements can be grouped to forms and 2) ontological reflection on role types will allow the presentation of players in a select box, a control that is more widely known than the current completion widget. Both planned features will help Topincs users to create more homogeneous topic maps that allow programmatic processing with less effort and without specifying a schema.

References

1. Ahmed, K.: TMSHare – Topic Map Fragment Exchange In a Peer-To-Peer Application. In: Proceedings of XML Europe 2002, (2003).
2. Barta, R.: Knowledge-Oriented Middleware Using Topic Maps. In: Maicher, L.; Garshol, L.M.: Proceedings of the Third International Conference on Topic Maps Research and Applications (TMRA'07), Leipzig; Springer LNAI 4999, (2008).
3. Bonifacio, M.; Bouquet, P.; Cuel, R.: Knowledge Nodes: the Building Blocks of a Distributed Approach to Knowledge Management. In: Journal of Universal Computer Science 8(6):191-200, (2002).
4. Bush, V.: As we may think. The Atlantic Monthly, July 1945. Reprinted in: Interactions (III) 2: 35-46, (1996).
5. Cerny, R.: Topincs – A RESTful Web Service Interface For Topic Maps. In: Maicher, L.; Sigel, A.; Garshol, L.M.: Proceedings of the Second International Conference on Topic Maps Research and Applications (TMRA'06), Leipzig; Springer LNAI 4438, (2007).
6. Cerny, R.: Topincs Wiki – A Topic Maps Powered Wiki. In: Maicher, L.; Garshol, L.M.: Proceedings of the Third International Conference on Topic Maps Research and Applications (TMRA'07), Leipzig; Springer LNAI 4999, (2008).
7. ISO/IEC IS, 0-2:2006: Topic Maps – Data Model, International Organization for Standardization, Geneva, Switzerland (2006).
<http://www.isotopicmaps.org/sam/sam-model/>
8. Maicher, L.: Topic Map Exchange in the Absence of Shared Vocabularies. In: Maicher, L.; Park, J.: Proceedings of the First International Workshop on Topic Maps Research and Applications (TMRA'05), Leipzig; Springer LNAI 3873, (2006).
9. Nelson, T. H.: Literary Machines. Sausalito; Mindful Press, (1981).
10. Sigel, A.: Report on the Open Space Sessions. In: Maicher, L.; Park J.: Proceedings of the First International Workshop on Topic Maps Research and Applications (TMRA'05), Leipzig; Springer LNAI 3873, (2006).

SocioTM – Relevancies, Collaboration, and Socio-knowledge in Topic Maps

Sasha Rudan¹ and Sinisha Rudan²

¹ HeadWare Solutions, Norway/Serbia, sasa.rudan [g-m-a-i-l]

² Magic Wand Solutions, Serbia, sinisa.rudan [g-m-a-i-l]

Abstract. Topic Maps (*TM*) standard solved a lot of problems in the information overload. With a semantic layer on the top of the existing data pools, TMs provide information interpretation and organization. However, user interaction with technology is still undeveloped and too explicit. This paper introduces *SocioTM* model; an extension of TM paradigm that includes relevancies, collaboration, and *socio-knowledge* (user-specific knowledge/ behaviors). Paper goes through relevancies implementation in SocioTM; relevancies building and population; relevancies interpretation, presentation; and navigation through **SocioTM**. Relevancies are introduced both on topic/ontology level and information (occurrences) level. Paper concludes with collaboration involvement in *SocioTM* building and with migration of socio-knowledge.

Keywords: Topic Maps, relevancies, semantic, ranking, rating, quality, visualization, voting, collaboration, groups, profiles, socio-knowledge, SocioTM, relieving, socio-potential-low, mountain-view, migration

1 Introduction

Topic Maps as a knowledge building and organizing technology is a fairly enough mature and powerful technology. We believe that our research should shift more to design of user interaction with technology; to make it more natural and what is the most important; more implicit. This implies need for components that can monitor and identify user behavior and preferences and be able to migrate it to the another knowledge pool. We would also like to introduce to TM arena a more native support for concepts already exploited in collaborative systems.

TMs generate two problems: *knowledge generalization* and *knowledge redundancy*¹. Knowledge generalization means that there is no any uniqueness in knowledge representation related to the specific user. Knowledge redundancy is introduced when each user/group wants to have separate knowledge representation (meta-data set) to identify their unique knowledge interpretation. There is a strong need for one unified TM set, but also for keeping personal uniqueness of every user/group.

In order to solve these important problems “***SocioTM HyperReliefing***” (***SocioTMHR***) or just shorter ***SocioTM²*** model is proposed (Figure 1.). The very name of the model makes two important implications: 1) model is intended to be gateway between proprietary applications and TM to be as much as possible integration transparent³ 2) model should be understood as an integrate part of TM; both in the way of necessity for it and in the relation to knowledge integration and migration of accumulated knowledge. Some features might be simulated using TCML or socio-ontology but there is a need for a standardized and in-box solution. We also need more researches and more generally accepted paradigms in that area.

¹ More general speaking; meta-data generalization and meta-data redundancy

² More details at www.SocioTM.org

³ Some system’s functionality (not being a part of TM standard) should be accessed through the separate API

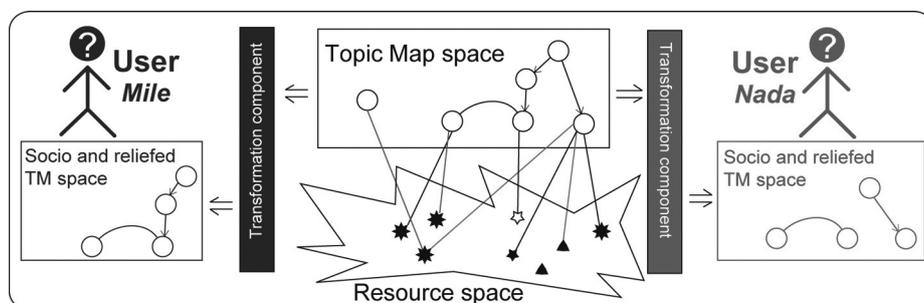


Fig. 1. Global overview of the SocioTM model. Different users are presented with different, profiled TM space (with a benefit to collaborative work and impact to the global knowledge)

SocioTM model gives relevancies to each TM-element (topic, association, class, occurrence, etc)⁴. SocioTM model makes us possible creation of relief and fuzzy representation of the knowledge. In this way, much easier usage of information and much better knowledge structure overview is made possible.

After introduction in Chapter 1, Chapter 2 presents relevancies population and creation; Chapter 3 talks about relevancies evolution; Chapter 4 presents all aspects of SocioTM interpretation; Chapter 5 gives a fast overview of SocioTM presentation; followed with Chapter 6 which presents navigation through SocioTM; Chapter 7 is about collaboration within SocioTM and migration of socio-knowledge and finally; Chapter 8 is just an overview of SocioTM implementation with; Chapter 9 as a conclusion of the paper.

1.1 Current state

Topic Map standard introduced roles, associations, scopes, themes, but no mechanism for easily ranking either topics or occurrences. Scope-concept and association-concept are binary-like concepts and more often part of ontology space (hardcoded) than user space. There is a need for more fuzzy and general concept.

To avoid information glut users are interested in browsing on meta-data level wanting to know which topics are more relevant, which path through the topic space will be faster and more effective. That is why users need to be presented with relevancies both on the occurrence and meta-data level and also with each TM-element.

⁴ If not explicitly noticed, this research is referring to all kind of TM-elements in general

Here is an illustration of the usage of SocioTM model; users may want to know which Mozart's compositions (topics) are the most popular. Moreover, for composer Marc-Antoine Charpentier they may want to know if he was much less outstanding composer than his the best known piece - *Eurovision* opening hymn.

Recommender systems as ubiquity phenomena and ranking algorithms are already highly researched ([Geroimenko2006], [Soboroff2009]) and this paper will not try to go deeper in that direction. It is up to **SocioTM** developer to choose the most appropriate models and algorithms.

1.2 Problem setting

This paper is a part of research on the system called *KnAlledge*⁵ being developed by *HeadWare Solutions*⁶ and *Knowledge Federation*⁷.

Our research is set in the following context: 1) resource and meta-data space are huge and highly interconnected; 2) interconnections are important for user; 3) no real-time response is required for new knowledge entrance; 4) there are many users that want to get suggestions about presented knowledge; to get structural concept of knowledge they are facing with; 5) they would like to be able to affect knowledge structure locally and preferably even globally; and 6) to be able to migrate with aggregated social-knowledge. Users want to start with preset world (not with tabula-rasa) and then to keep personal memories and make global impact onto that world.

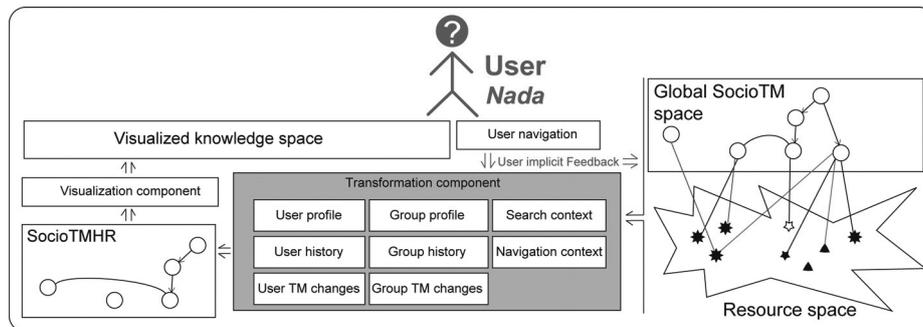


Fig. 2. SocioTM system detailed.

⁵ KnAlledge system is about collaborative knowledge generation, knowledge merging and unified presentation of the whole content (referring to the same topic) as the one filtered and merged content. More details at www.KnAlledge.com or www.Memepolis.com

⁶ More details at www.HeadWareSolutions.com

⁷ More details at www.KnowledgeFederation.org

2 Relevancies population and creation

This chapter introduces a mechanism for adding relevancies to TM-elements. TM-element population does not have a problem with meta-data-entrance-laziness. Opposite to occurrence-relevance space where we are not aware of user satisfaction with occurrence quality, in TM-element-relevance space we have good methods to monitor user satisfaction. For example, if user navigated in one direction we know that path was good⁸. The same is about a topic; if user accessed some resource which is an occurrence of that topic, etc.

SocioTM assumes static-relevancies (*static-SocioTM*) and dynamic-relevancies (*dynamic-SocioTM*). Static-SocioTM contains *static relevancies* that are persistent over successive use of Topic Map. On the other side dynamic-SocioTM contains *dynamic relevancies* that are being calculated from the static ones intended to present user/search/navigation specific scope of SocioTM.

2.1. Implementation

The easiest way to implement relevancies is by adding weight to each TM-element (similar to weighted graph or ANN (*Artificial Neural Network*) topology). However, better way would be if we could also store *socio-knowledge* (knowledge related to user/group), in a form of already extracted rules (behaviors, preferences, etc) accompanied with accumulated user's actions that are waiting to be processed. Later, accumulated actions could be used for relearning; modifying existing and creating new rules. In this way, we can predict and suggest user's actions and interest but also manage user explicit needs (like ranking specific topic).

Another solution would be to create a special storage (i.e. TM storage) for storing socio-knowledge. In this way we will have users', groups', and global SocioTM. This solution will be elaborated in this paper. Final draft for the socio-knowledge storage's (*SKS*) taxonomy will be provided on the project's web portal.

We also have to balance both with user privacy and collaboration goals. In order to keep user privacy, it is possible either to immediately populate global/group SKS to the response of user activities or to try to extract knowledge and generalize it. In both cases we are using user personal data only at the moment they are already going through the system, so the user is less concerned with the privacy aspect. However, real-time algorithms needed for real-time processing of

⁸ Relevance of association in relation to the source topic, not a relevance of destination topic

the user activities are both more difficult to develop, and more computational expensive [Linden2003].

For extracting knowledge/patterns and recognizing user behavior, we can use (i.e. ANN) (un)supervised learning methods [Anderson1992]. It is important to recognize users' behaviors in navigation and browsing, but also interests in more general concepts; like specific scopes, topic/occurrence, and association classes. It is also important to recognize users' preferences relating to the way they anticipate data presentation and learn; linear, spiral, etc. In this way system can predict user's navigation, interests and presentation preferable. Opposite to that would be a simple monitoring and recording user actions and then promoting them globally.

User interest in some element needs extremely complicated analysis including lexical understanding of TM content/resources, and it is a part of another, later research.

2.2 User implicit feedback

This is an important aspect of the system. It provides a chance that user's actions permanently changes original TM. In this way we both integrate collected knowledge with information-pool and provide a new user (which does not belong to any group or has own profile) with a chance to use already customized and evaluated knowledge. Some kind of feedback delay and feedback evaluation helps us in providing more globally-approved knowledge.

3 Relevancies evolution

This chapter presents a way of evaluating acquired socio-knowledge to be ready for later interpretation. Relevance evolution is more an offline process compared to the relevancies interpretation.

Opposite to almost real-time response of the new content [Das2007] our solution is more likely to get precise and highly evaluated answers with possibility of offline calculation [Linden2003].

By creating *clusters* of users (using clustering or other unsupervised learning algorithm) we can greatly reduce computation space, and reduce complexity of algorithm from $O(M*N)$ to $O(M+N)$ or even less⁹.

⁹ M presents the number of users and N presents the number of TM-elements

The challenging problem of populating user profile can be overridden by a clustering concept. Our idea is very simple; the starting assumption is that every cluster will have at least one user willing to populate questioner and to build user-profile. On the other side all users from one particular cluster can be identified with cluster-behavior (average of all populated profiles in that cluster); after normalizing cluster-behavior with user-specific behavior. We could even inspire user to confirm/reconsider automatically assigned user-profile; preferable to readjust it.

An important phenomenon developers have to pay attention on is the phenomenon of *promoted elements*. Imagine one element that users gave impression about. Users' reaction will stimulate other users to do the same. For example, if users followed association A_k from topic T_m , another user would probably unjustifiably take suggested A_k when navigating from T_m . This creates avalanche effect. Promoted elements can be handled in the following ways: 1) initial popularity divided with frequency of use, 2) postpone popularity propagation, or 3) evaluate if user was satisfied with suggestion. All these 3 approaches can be combined.

4 SocioTM interpretation

This chapter explains a transformation of static-TM with static relevancies into dynamic-TM with dynamic relevancies. In other words it explains transformation from *collective knowledge* to *scoped/profiled knowledge* (i.e. user's SocioTM). Let us just note here that static-TM is not static in a general meaning. It also evaluates through user-feedback (in this way underlining importance of collective knowledge), by growing population and by changing/voting/authorizing user-preferences.

$$TM_{\text{DYNAMIC}} = f_{\text{USER-NORMALIZING}}(TM_{\text{STATIC}}) \quad (1)$$

Figure 4 overviews a process of initializing user's SocioTM and process of static/dynamic normalizing activities during interaction with SocioTM.

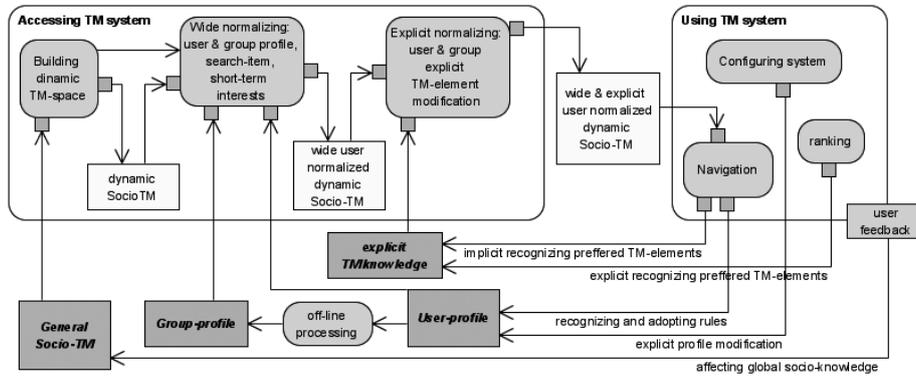


Fig. 4. SocioTM normalizing activities

4.1 Building dynamic SocioTM

There are two major models in building SocioTM: 1) duplicating static SocioTM into dynamic one; 2) generating dynamic SocioTM from the static one *on-the-fly*. No matter which model is used, it will be referred as *dynamic-SocioTM* for the sake of clarity.

4.1.1 Building dynamic SocioTM as a copy of topic space

This model makes a copy of static SocioTM in which all transformations are performed. It is completely safe to make changes against it and algorithms seem to be more efficient and easy to implement. However, for pattern recognition, the next model seems to be more practical, so it should be partially used.

4.1.2 Building dynamic SocioTM on-the-fly

This model does not create a copy, but introduces *mapping-layer* responsible for mapping static SocioTM into dynamic one on-the-fly. This model is more implementation-demanding, but on the other side it is more careful with memory consumption and it gives nice possibility of *user-feedback* implementation.

4.2 Wide normalization

Wide normalizations are all SocioTM normalization activities that have effect on whole SocioTM space and not only on the specific TM-elements.

4.2.1 Normalizing TM with user profile

Each user has a personal profile which represents user's behavior. Personal profile is built and profiled over time, by monitoring user's behavior/interests or by manual user intervention. As we already mentioned, initial user profile is cluster profile.

User profile contains explicit user set of preferences and expectations. It also contains a set of rules learned by (un)supervised learning. Rules can also be offered to user afterwards to fit them more precisely and to be stimulated to create new rules.

4.2.2 Long-term and short-term user interests

Every user has long-term and short-term interests. Long-term interests are recognized through user manual profiling or by monitoring user's behavior over a period of time. However, by avoiding short-term interests we are attracting user's present interests into wrong direction, driven by long-term-interests' suggestion.

4.2.3 Normalizing TM with search-item

Search-item contains in itself a lot of filtering information to provide not only result but also to generate separate *view* on TM. In practice, it is done by normalizing and filtering all TM-elements in TM according to search-item and user profile. How much search-item can help, depends on search-item semantic richness. One important note is that search-item is not only about 1) normalizing SocioTM, but also about 2) cutting-off non-relevant parts of SocioTM space.

4.3 Explicit normalizing with user explicit-socio-knowledge

Explicit-socio-knowledge presents a set of user explicitly modified TM-elements. There is one-to-one association between each record in explicit-socio-knowledge and addressed TM-element. Process of normalization consists of iterating through all records and appropriate modifying every addressed TM-element.

4.4 Normalizing TM through user navigation and time

Time and navigation is a very reach source of implicit knowledge retrieval. As we will see later, by only monitoring user navigation through SocioTM, system is able to *implicitly* recognize user behavior, relevancies and expectations.

4.5 Conclusion

The final normalizing function (summing all normalizing activities) looks like:

$$f_{\text{USER-NORMALIZING}} = f_{\text{GROUP-PROFILE}} \times f_{\text{USER-PROFILE}} \times f_{\text{ST/LT-INTERESTS}} \times f_{\text{SEARCH-ITEM}} \times f_{\text{EXPLICIT-SOCIO-KNOWLEDGE}} \times f_{\text{NAVIGATION}} \times f_{\text{TIME}} \quad (2)$$

5 SocioTM presentation

5.1 Challenges with Topic Maps presentation

Even if user navigates through meta-data space there is still a huge overload of meta-data at that level but also overload of knowledge in general. This means that our system still have to cope with the problem of visualization/presentation. Relevancies introduction is a try of avoiding that problem, but it introduces new challenges; *view-clipping* and *presenting* the SocioTM.

5.2 View-clipping

When user is browsing SocioTM user should be presented with limited knowledge. The best way is view-clipping related to user tuned *relevance-threshold*. Clipping should go both horizontal and vertical. *Horizontal-clipping* means clipping to the knowledge relevant to the present user interest. If user approaches knowledge border, *socio-potential law* will extend the knowledge in the way it is presented in the next chapter. *Vertical-clipping* includes clipping by the *relevance-threshold* and clipping by the *knowledge-abstraction-level* user is interested at the moment.

5.3 Mountain-View paradigm

Mountain-view paradigm is related to the way of presenting data to the user [Karabeg2002]. Main idea is to use the user’s best orientation tool; spatial and time orientation to understand knowledge structure and browse through it. As we will see in the next chapter, relief will continue to change through user navigation, new peaks will appear, and old disappear. Mountain-view paradigm provides user with visual interpretation of knowledge structure stored in SocioTM and it changes with change of user’s interests and with user navigation through SocioTM.

6 SocioTM Navigation (socio-potential-law)

There are almost no researches in the area of recommenders, referring to the way of navigating through *data-set* (in our case SocioTM) [Geroimenko206]. The most of them are related to the way of recommending items (in our case TM-elements). In the info glut recommendations/relevancies are also needed for navigation paths.

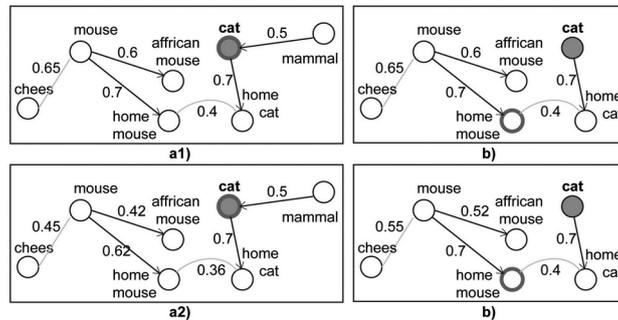


Fig. 5. Socio-potential-law

After dynamic-SocioTM is created SocioTM model can work in the *spatial-time* relevance-domain which means that relevancies are being evaluated and changed over the time and by user-navigation through dynamic-SocioTM. In this domain it is used something we call *socio-potential-law*. The socio-potential-law works similar to the physical force-potential-law¹⁰; all relevancies are decreasing weighted relatively to the present force-center/epicenter. Additional tensions could be introduced, like search-item origin, etc (Figure 5).

¹⁰ It falls in the group of easily convergent Force-based algorithms [Fruchterman1991]

As an implication this law gives us possibility to evaluate relevancies spatially and over time. Spatiality is evaluated by implementing some of the appropriate metric [Bruls2000].

On Figure 5.a1) someone can see that for search-item "cat" (blue circle) dynamic-relevance of every TM-element can calculate by superposition of cumulative metric distances (weights) on path from epicenter to the observed TM-element normalized with static-relevance of the observed TM-element. This makes possible transposition metric to the vertical dimension (*Mountain-View*).

Introduction of additional epicenters gives precedence to other user tensions; like search-item (blue-circle) and user's present position in SocioTM (red-ring); Figure 5.a2). Through navigation through the SocioTM user relocates her/his tensions and some other topics become more important (Figure 5.b1 and 5.b2).

Just this evaluation of relevancies through spatial-time dimension gives us a chance to make a user-feedback to SocioTM. This is an exciting area for the future research.

Another fascinating manifest we see here is the following: if we have a bare info-pool without any recommendations or relevancies, we can just let users navigate through it; probably with some support of lexical-similarity-recommenders and lexical/tag metrics. Without forcing users to make any explicit recommendations we still can collect amazingly rich cognition about knowledge relations, contexts, relevancies, etc. We believe this area opens us a new horizon of researches in implicit social-knowledge population.

7 Collaboration within Topic Maps

Even some experts debate about meta-data overload; meta-data cannot be overload since they are supposed to help better navigation and filtering information. This means that user does not have to see meta-data but only to use them.

With topic maps and similar technologies we are providing user to navigate/browse/view not only on information level but also on meta-data level. This makes us responsible and concerned about meta-data overload. Our belief is that information overload is not about data itself, but about information presentation and providing information consumer with ability to get overall picture of data and main concepts of knowledge stored in that information pool.

SocioTM provides a better overview and knowledge selection, but at the same time it solves a big collaborative issue by sharing user experience and keeping individual aspect at the same time. In that way, we avoided duplicated socio-

semantic space (meta-data redundancy) and also afforded user specific behaviors and expectations (we avoided meta-data generalization).

7.1 Socio-knowledge migration

Very important feature of SocioTM is having social-knowledge, user-profiles, and relevancies separated of TM content. The reason is simple, knowledge about knowledge, and meta-data in general should be reusable and therefore be possible to migrate to the other information pool.

That was the primary reason for introducing separate socio-knowledge storage (SKS) within our system. This makes possible mapping aggregated socio-knowledge to the other TM.

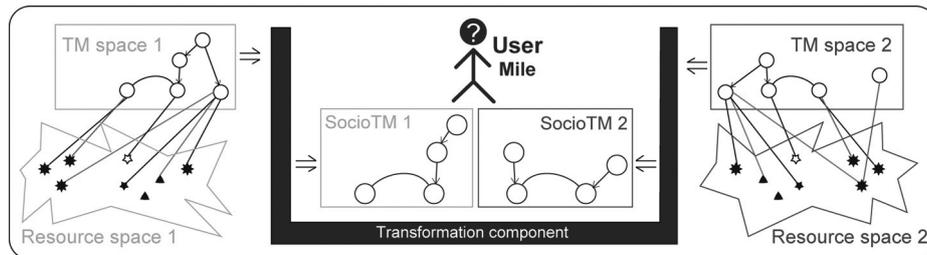


Fig.3 Migration of socio-knowledge

To make social-knowledge migration more efficient there is a need for PRIs/PSIs (Published Subject Identifiers) [Pepper 2008] to map the socio-knowledge. We would like to introduce PRI concept not only at the topic level, but also at the ontology level (which we believe should be easier to negotiate about.)

8 SocioTM implementation

SocioTM taxonomy presented here is just a glimpse of the final draft¹¹

- Topic classes: `topic_visited`, `topic_ranking`, `topic_examined`, `association_ranking`, `association_followed`, etc
- Association classes: `topics_related`, etc
- Occurrence classes: `relevance_value`, `visiting_frequency`, etc

¹¹ Final draft would be presented on the SocioTM portal (<http://www.sociotm.org>)

8.1 Introducing SocioTM into the existing systems

Vertical compatibility is always challenging with introduction of new concepts and technologies. If system is build modular than introducing SocioTM should not be a dramatic issue¹². We are open to the other researchers and developers for possible challenges and help in system modeling.

9 Conclusion

In this paper we presented extension to the Topic Map standard (general enough to be extend to the similar technology) that supports socio-knowledge added on the top of classical TM providing more structural knowledge and knowledge profiled to the user, but also collaborative to the community.

When it comes to standardization problem, we believe that standardization is extremely important to make this concept native and permanent companion of TM.

We can imagine different experts providing their overview/knowledge interpretation to the audience. Users would be able to choose either one or another expert (i.e. music expert) to follow her/his interpretation. In this case we would be able provide on-line, dynamic and living books about the same area and with similar content (the same global and enormous Topic Map) interpreted in a different way.

The new challenge would be to add a contextual relieving not only to the user-specific-context but also to the search-item-specific-context. At the present moment we see it as memory-demanding issue without easily generalization/pattern recognition approach so we leave it for the later research.

References

1. Karabeg, D.: Designing Information Design. Information Design Journal Vol 11.1, pp. 82-90 (2002).
2. Soboroff I., Nicholas C., "Combining Content and Collaboration in Text Filtering", IJCAI'99 Workshop on Machine Learning for Information Filtering, Sweden (1999)
3. Geroimenko V., Chen C., "Visualizing the Semantic Web: XML-based Internet and Information Visualization", (2006) 102-123

¹² Especially if we are using only functionality presented through standard TM gateway

4. Rudan S., "Semantic enrichment of multimedia information and processes", 2005
http://www.sinisarudan.com/computer_sciences/multimedia&semantics/SemanticEnrichmentMM.pdf
5. Lachica R., Karabeg D., Rudan S., "Quality, Relevance and Importance in Information Retrieval with Fuzzy Semantic Networks", TMRA, Germany (2008)
6. Linden G., Smith B., and York J., "Amazon.com Recommendations; Item-to-Item Collaborative Filtering", IEEE Internet Computing, Volume 7, Issue 1 (January 2003) 76-80
7. Das A., Datar M., Garg A., Rajaram S., "Google News Personalization: Scalable Online Collaborative Filtering", Industrial Practice and Experience, Alberta (2007)
8. Bruls M., "Squarified treemaps", TCVG Symposium on Visualization, (2000) 33-42
9. O'Connor M., Cosley D., "PolyLens: a recommender system for groups of users. Proceeding of the ECSW 2001, Germany (2001)
10. Pampalk E., Goto M., "MusicRainbow: A New User Interface to Discover Artists Using Audio-based Similarity and Web-based Labeling"
11. Sabre J., "'Relevance' in Information Retrieval", 2004
12. Gupta S., Nenkova A., Jurafsky D., "Measuring Importance and Query Relevance in Topic-focused Multi-document Summarization"
13. Anderson D., McNeill G., "Artificial Neural Networks Technology",
http://www.SocioTM.org/docs/ANN_Technology.pdf, 1992
14. Fruchterman, T. M. J., Reingold, E. M., "Graph Drawing by Force-directed Placement", Software: Practice and Experience, 21(11), 1991
15. Pepper S., "Global data identifier", Published Subjects and global identifiers, Oslo (2008)

The Effects of Topic Map Components on Serendipitous Information Retrieval

Sunmin Won¹ and Sam Gyun Oh²

¹Sungkyunkwan University, Department of Library & Information Science
won.sunmin@gmail.com

²Sungkyunkwan University, Professor, Department of Library & Information Science.
samoh@skku.edu

Abstract. Some recent studies on information retrieval and information seeking have examined the utility of serendipitous discovery. This research argues that serendipitous discovery has a positive impact on information retrieval and can happen most frequently in semantic web built on the framework of topic maps. This paper discusses the components of topic map that influence serendipitous discovery as well as the elements of topic map designs that may enhance serendipitous discovery. To that end the results of a study on the effects of serendipitous discovery in topic-map-based ontology systems are discussed in the context of information seeking.

1 Introduction

Fast and easy access to massive and various resources are now available to information users. Yet finding appropriate information that meets users' needs has become an increasingly difficult task. While the challenges multiply with polysemous words and hard-to-define needs of information users, current web technologies primarily support keyword-based searches.

The semantic web environment seeks to enable machines to understand and process information, allowing users to better obtain relevant information. One of the latest concerns among researchers is the utility of serendipity in information retrieval (IR) and information seeking. Russell(2007) defines serendipity as fortunate discoveries made by accidents and as a scientific procedure[1]. With

well-organized data the potential for web encounters of a vast number of interesting materials increases.

The topic map, which is one of the frameworks for building the semantic web, provides capability to connect all semantically-related information. Therefore it enables human judgment and decision in relation to relevant information, and enhances serendipity through systematic facilitation of users' chance encounters with information.

The current paper lists suggestions for the design of the topic map that facilitates serendipitous information retrieval. To that end, data from a qualitative study in the context of serendipity which examined the effects of topic map-based ontology system on the information needs of information seekers was analyzed. The study consisted of observations and interviews of 10 college students.

2 Serendipity and Information Retrieval

2.1 The Concept of Serendipity in IR

Serendipity is defined as a situation in which one stumbles upon appropriate information. While information acquisition commonly involves the use of a search/query mechanism or browsing/scanning, it is believed that the latter is more inductive to serendipity.

There are a number of conceptual studies on serendipity. Erdelez (1997), for instance, identified a "serendipity-prone" group among some academic library users who reported "the pressure of the abundance of information waiting to be encountered". Ross(1999) interviewed 194 committed readers and found that these readers were "finding without seeking", gaining valuable insights from their pleasure reading of materials that interested them. Williamson(1998) also discovered incidental information acquisitions among 202 Australian senior citizens from their phone conversations. These studies support that there exists certain dynamics between serendipitous discovery and individuals [3].

2.2 The Usefulness of Serendipity in IR

What makes serendipity useful in IR is its implications for opportunities to find information that is either potentially valuable or capable of generating new knowledge. Some research has found serendipitous discovery as a common

experience among information users and as one that assists their information seeking tasks.

Toms(2000) reported an experiment in which the effects of unintended information retrieval were studied using a specially developed information retrieval application. Results showed that the information obtained through serendipitous discovery was thought to be more meaningful than the information obtained through keyword searching method. Furthermore, information seekers were more involved with retrieval that was inductive to serendipitous discovery. Allen E. Foster and Nigel Ford(2003)'s qualitative study of serendipitous discovery among inter-disciplinary researchers revealed that such discovery was broadly experienced and was likely to provide information relevant to their research process. They also found that certain attitudes and strategic decisions were perceived to be effective in exploiting serendipity when it occurred.

3 Serendipity in Topic Map

3.1 Overview of Current Research

The initial purpose of this study was to analyze the impact that serendipity, which is caused by a topic-map-based ontology system, brings to the clarification of users' information needs. For this purpose a topic-map-based information retrieval system was developed using the ontopoly editor and was given the name TIRS(Topic Map Information Retrieval System). TIRS provided information on every nation of the world as well as on the top 100 global corporations. It had 1,372 topics, 3,924 associations and 4,865 occurrences. Ten participants searched information with TIRS in a largely uncontrolled environment. The only constraint the participants had to follow was that they employ only browsing and scanning as their search strategy. Data was collected by qualitative methods of observations, interviews and a 'Think Aloud' technique which recorded the participants' thoughts during the entire search process. Then unitization and categorization of the data was done, followed by data analyses.

3.2. Findings

The data from the study revealed the following:

- a) The participants experienced serendipitous discovery during information retrieval and were able to locate relevant information with ease using the TIRS.

- b) The participants either adopted the serendipitous discovery of information or utilized it to modify their information needs and retrieval strategies. Modifications included more specific information needs and wording as well as the addition of words and narrowing down of retrieval ranges.
- c) Topic-map-induced serendipity was an element that influenced the changes in the types of information that the participants felt that they needed.

4. TAO and Serendipity

The present study examined how the information seekers utilized the components of the topic map. It also identified the components that induced most serendipitous discoveries.

4.1. Use of the Topic Map Components

As is summarized in Table 1., the component that yielded the highest percentage of usage(40.8%) was the "topic type". The participants employed this component primarily to access individual information (29.5% of all cited purposes), but relied on it as well when checking and searching the information that belonged to super-class categories.

Next in usage was the "association type", which yielded 32.0% of the frequencies. The participants employed this component primarily to check information by its associative relationships (16.3% of all cited purposes), and to identify information relevant to their known needs rather than to search for one that might generate needs.

Lastly the "occurrence type", which accounted for 27.3% of the usage, was most applicable when the participants wanted to access detailed information; 11.8% of all cited purposes fell into this category. The "occurrence type" was also useful when searching for the value of a specific item or checking the overall content of a specific item. Incidentally this component was cited as particularly helpful when checking ranks.

Nevertheless it is important to note that during the face-to-face interviews administered at completion of retrieval sessions six participants cited the "association type" as the most frequently used component, while only one participant mentioned the "topic type" as such. An explanation for the discrepancies may lie in the degrees of actual satisfaction: The information

obtained or discovered by association type may have been far more effective in meeting the information needs of these participants.

Component name	Purpose of using	The Number of times	Rate	Total	Rate
Topic Type	- Accessing to individual information	118	29.5%	163	40.8%
	- Holistic searching in super ordinate category	45	11.3%		
Association Type	- Related information encountering	61	15.3%	128	32.0%
	- Encountering information what generate new information need	2	0.5%		
	- Browsing organized information what relation based	65	16.3%		
Occurrence Type	- Holistic searching in super ordinate category	37	9.3%	109	27.3%
	- Related information encountering	19	48.0%		
	- Rank checking	6	1.5%		
	- Specific information checking	47	11.8%		
Total		400	100%	400	100%

Table 1. Rate of using of Topic Map component during information seeking behavior

4.2. Topic Map Components Inducive to Serendipitous Discoveries

Valuable insight into what components may best enhance serendipitous discoveries can be gained from the data from face-to-face interviews. In order to identify these components, cases where the participants recognized serendipitous discoveries either instantly or after they had initiated modifications on their information needs were carefully followed up. Results confirmed that the "association type", which produced 17 cases of serendipitous discoveries, had been the most effective component. Interestingly the component "topic type"

produced no more than a single reported case, while "occurrence type" was cited in 12 cases. Related statistics are summarized in Table 2.

Component name	The number of time what inducing serendipitous situation	Rate
Topic Type	1	3.3%
Association Type	17	56.7%
Occurrence Type	12	40.0%
Total	30	100%

Table 2. Rate of Topic Map component what inducing serendipitous discovery situation

Despite the fact that it accounted for 40.8% of the usage, only 3.3% of the serendipitous discoveries was linked to the "topic type". In comparison, the components "association type" and "occurrence type" corresponded to 56.7% and 40.0% of the serendipitous discoveries, respectively. In all, serendipitous discoveries were largely unpredicted by the mere frequencies of the usage of the components; what seems more crucial is the degrees in which the participants' information needs had been met by these components.

As indicated in Table 1, the participants used the components mostly to browse information of relationship-based presentations in the "association type", and to check specific information in the "occurrence type". That is, in terms of serendipitous discoveries, information organized by relationships and the discovery of specific information seemed to have been the effective inducer for each type. Results from further interviews with the participants confirmed this speculation. They noted that the strength of the "association type" lies in its grouping of information by relationships and its exhaustive displays of related information for any given selections. In the "occurrence type" they valued the presentation of information in descending/ascending orders as well as the inclusion of specific values and substance as these features enabled them to compare different pieces of information.

5 Suggestions from Participants

During the course of the in-depth interviews the participants discussed the strengths of the topic map system and expressed ideas that might enhance serendipity.

5.1 Association Type as Component: Strengths and Desired Improvements

A. Utility of Relationships

The participants reported that the association type proved convenient in that it allowed searches, in a chain fashion, of directly related information. High level of relatedness among the pieces of information also meant fewer efforts to filter out unnecessary information on their part.

B. Reduction in Comparisons between Information

Information was arranged using diverse relationships among different topic types, making it possible for the participants to peruse it at one glance. The participants were of the opinion that their searches had been made more economical and less cumbersome by these arrangements. However, they also requested that the singular criterion of descending/ascending order of arrangements be expanded upon. They said, for instance, that different pieces of information could be displayed according to their relatedness to another piece of information.

C. Semantic Recognition

The participants judged that the association type facilitated their searches as it was expressive enough of the contents and the types of relationships in the information. Such semantic representations gave them the necessary first clues. With the relatively easy guessing work on what each piece of the information was about, the participants successfully managed to narrow down their searches.

On the other hand, the participants also noted that ambiguous expressions only served as deterrents since these expressions caused them to guess wrong. Consequently, they suggested that the association type names be more explanatory and analogical in relation to the contents and relationships of the information, .

5.2 Occurrence Type as Component: Strengths and Desired Improvements

A. Presentation of Details

The participants agreed that the at-a-glance display of numerical values and specifics on contents is the strength of the occurrence type. They also appreciated the fact that direct links to the topic types were a part of this display. These features of the occurrence type facilitated searches for information with concrete

data, and at times provided clues for narrowing down the categories of searches. Yet the participants mentioned that grouping numerical values and specifics on contents according to the relatedness among different pieces of information would be a welcome option to improve the functionality of the occurrence type.

6 Suggestions for Topic Map Designs to Enhance Serendipity

On the basis of the quantitative data and the results from the interviews some considerations for topic map designs for better serendipity follow.

6.1. Organization of Information

Classification by relationships, which renders relative ease in locating relevant information, was the element that made topic map based IRS convenient. Maximizing the use of the filtering function of the topic map is crucial in the expression of the distinct relationships among different pieces of information. Presenting these pieces of information under similar or related categories is also necessary.

"Association" and "association type" provide the most basic filtering function of the topic map, while "hierarchy", which represents up and down relationships, is most suitable for displays of information by similar or related categories. "Scope" and "role type", which are not directly shown to users yet serve as the cornerstone for limiting ranges and clarifying relationships among information, are two other components that must be fully utilized.

6.2. Semantic Labeling

Explicitness in the expressions of the relationships between different pieces of information is a key element. That is, labeling of the association should be made sufficiently explanatory of the relationships between topics. If the related topics are represented in the label itself, it would increase the efficiency and effectiveness of the association type.

The data from the present study were drawn from undergraduate students in Korea using the Korean language as the medium of communication. Consequently the implications of the study may not be applicable to all populations and languages. Much more linguistic research is required to clarify issues in labeling.

Nevertheless certain aspects of labeling that emerged from the present study do deserve a close attention. In general, when the understanding of the topic map is less than complete, labeling in and by itself does not sufficiently express the relationships between topics. Furthermore, one label cannot represent all relationships: One must employ different grammar if, for instance, the position of the subject and the predicate is reversed.

6.3. Organization of Displays

The participants agreed that the most prominent strength of the topic map based IRS is its capacity to provide information that has been arranged by relationships. Organized information meant better access to relevant information. Yet the participants also noted that the IRS does not go beyond the level of showing the relationships between two directly involved topics. A desirable enhancement, they said, would be the addition of a second-level organization by which other related information is displayed as well. Even though such expressions exceed the basic functions of the topic map, they are still feasible in its construction with query languages such as TOLOG or TMQL.

7. Conclusion

The participants of the present study highly valued the convenience and utility of the topic map. The majority of these participants experienced serendipitous discoveries and judged them to be an integral of information retrieval.

The topic type was the most-used, yet the least-productive of all topic map components as it was connected to but a single case of serendipity. The participants used this component primarily to access individual information. There may be various features that trigger serendipitous discoveries, but the data from the present study suggest that relatedness among information is an extremely important feature for such discoveries. This premise was supported by the fact that most participants reported serendipitous discoveries with the association type, and that their use of the association type increased more toward the end of their information retrieval sessions.

Effective and efficient serendipitous discoveries require maximum utilization of relatedness. This necessitates better organizations of information and enhanced semantic expressions as well as technical applications of TMQL or TOLOG. In all, what is fundamental to the design of topic map based IRS seems to be the crucial role that the association type may play for serendipity. The association

type and the association name should be explanatory and analogical if one is to reflect the well-defined and various relationships between the pieces of information (i.e. the topic types in some respect).

The present study produced a small quantity of data involving a small number of subjects. However, within the context of this experiment it revealed that topic map is inductive to serendipitous discoveries, and that these discoveries facilitated the participants' information retrieval, especially in relation to the clarification of their information needs.

References

- Allen Foster & Nigel Ford. 2003. Serendipity And Information Seeking: An Empirical Study. *Journal of Documentation* 59(3) : 321-340.
- A. Green. 1990. What do we mean by User Needs. *British Journal of Academic Librarianship* 5 : 65-78.
- Brenda Dervin. December 1998. Sense-making theory and practice: an overview of user interests in knowledge seeking and use. *Journal of Knowledge Management* 2(2) : 36-46
- Catherine C. Marshall & William Jones. January 2006. Keeping Encountered Information. *Communications of the ACM* 49(1) : 66-67.
- E. Toms. 2000. Serendipitous Information Retrieval. Proceedings of the First DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries, Zurich, Switzerland European Research Consortium for Informatics and Mathematics. [online] http://www.ercim.org/publication/ws-proceedings/DelNoe01/3_Toms.pdf
- Graham Moore. December 2000. Topic Map technology - the state of the art. XML 2000 Conference & Exposition, Washington, USA.
- Jürgen Beier. 2001. Navigation and interaction in medical knowledge spaces using topic maps. *International Congress Series* 1230 : 384-388.
- S. Erdelez. 1999. Information encountering: it's more than just bumping into information. *Bulletin of the American Society for Information Science* 25(3): 25-29.
- S. Pepper. 2002. The tao of topic maps : Finding the Way in the Age of Infoglut. *Ontopia*. [online]. <http://www.ontopia.net/topicmaps/materials/tao.html>
- S. Pepper, G. Moore. 2001. XML topic maps (XTM) 1.0. [online] <http://www.topicmaps.org/xtm/1.0/>

Poster Session

The Contributions for the Poster Session

Lutz Maicher¹ and Lars Marius Garshol²

¹Topic Maps Lab, University of Leipzig, Germany
maicher@informatik.uni-leipzig.de

²Bouvet AS, Oslo, Norway
larsga@bouvet.no

Abstract. This paper contains the abstracts of the posters which were presented at the TMRA 2008 conference.

Making Metadadate Alive: Migrating Metadata into Richer Semantic Relationships Using Topic Maps-based Ontology

Myongho Yi, School of Library and Information Studies, Texas Woman's University, P.O. Box 425438, Denton, TX 76204-5438, US,
topicmap@gmail.com

Due to the increasing amount and complexity of digital resources, there are several critical issues that arise in digital environments such as ill-structured and poor management of digital information. Different information organization approaches have been used to address these issues. In particular, Semantic Web has been explored for 10 years; however there are not many practical applications. This is in part due to the fact that much attention has been given to the creation rather than the migration of existing data. In addition, the lack of guidelines for choosing the right migration approach, whether Topic Maps or Resource Description Framework (RDF), needs to be addressed. This paper presents a comparison of Semantic Web Data Models (Topic Maps and RDF), followed by an example of migration of existing metadata into ontology-based data for Semantic Web.

RDF to Topic Map, Compatibilities and Differences in Design Process for Bam 3DCG Ontology

Elham Andaroodi, National Institute of Informatics, 2-1-2,
Hitotsubashi, Chiyoda-ku, Tokyo, Japan, elham@nii.ac.jp

Kinji Ono, National Institute of Informatics, 2-1-2, Hitotsubashi,
Chiyoda-ku, Tokyo, Japan, ono@nii.ac.jp

Motomu Naito, Knowledge Synergy, 3-747-4-203, Kusunokidai,
Tokorozawa, Saitama, Japan, motom@green.ocn.ne.jp

This poster addresses the compatibilities and differences between RDF and Topic Map for conceptualizing the knowledge of a metadata-based multilingual ontology, for the process and output of our research project related to a subset of cultural artifacts and world heritages. It starts with the architecture of the ontology and the design process as RDF consisting of 5 classes: Outline, Person, Research, Resource, and Outcome of the target domain. Here the subclasses are defined to complete the taxonomy and are given properties using the instances.

Later we compare different features provided by Topic Maps for conceptualizing the knowledge, for example topics with the taxonomy of classes in RDF and associations or occurrences with properties in RDF. It discusses further options in Topic Maps like “scope” or “association role type” that provide more varied alternatives for design of an ontology comparing to RDF. The focus of this comparison is multilingual typology schema of our ontology.

After the process of importing the RDF into Ontopoly Omnigator, we try to argue the shortcoming of RDF to Topic Map (RDF2TM) of Omnigator specially for representing the hierarchical schema in RDF. As each class in RDF is changed into a topic in TM regardless of its position in the hierarchy, the taxonomic characteristic of the ontology is lost. Besides each instance in RDF can be a topic in TM and be placed beside the classes. This probably makes a significant difference in conceptualizing a knowledge model using RDF or TM. We made this comparison using our metadata typology schema.

Moreover as Topic Map is an ISO standard a schema is available by default while a new file is created that makes different Topic Maps more coherent but limits the designer for modeling the desired schema. However Subject Identifier facilitates to map and merge different Topic Maps. Finally we conclude with our solution to reach to a proper XTM version of our ontology by adjustment and sometimes remodeling and introduces our implementation to develop topic map web applications by using Ontopia Navigator Framework which is included OKS (Ontopia Knowledge Suite)TM.

From Connexions Content to Content Connexions: Organizing Open Learning Resources with Topic Maps and XSLT

Lars Johnsen, University of Southern Denmark, Engstien 1, 6000 Kolding, Denmark, larsjo@sitkom.sdu.dk

Darina Dicheva, Winston-Salem State University, 601 Martin Luther King, J. Drive, Winston-Salem, NC 27110

Connexions is a project aimed at providing free and open content for educational purposes on a global scale. In this presentation it is discussed how information associated with open learning modules on the Connexions web site may be extracted and mapped onto topic maps. It is assumed that mapping Connexions content to topic maps will lead to improved navigation and more extensive reuse of the content. The focus of the presentation is on the proposed use of simple XSLT style sheets for the mapping and the mapping model itself. In addition, it is demonstrated how the educational topic map editor TM4L has been extended to support the generation of topic maps based on Connexions content using XSLT.

Use of Topic Maps to support Learning Organizational Memory

Adeline Leblanc, HEUDIASYC CNRS UMR 6599, Universite de Technologie de Compiègne, BP 20529, 60205 Compiègne CEDEX, France, fadeline.leblanc@utc.fr

Amjad Abou Assali, HEUDIASYC CNRS UMR 6599, Universite de Technologie de Compiègne, BP 20529, 60205 Compiègne CEDEX, France, amjad.abou-assali@utc.fr

Marie-Hélène Abel, HEUDIASYC CNRS UMR 6599, Universite de Technologie de Compiègne, BP 20529, 60205 Compiègne CEDEX, France, marie-helene.abel@utc.fr

Dominique Lenne, HEUDIASYC CNRS UMR 6599, Universite de Technologie de Compiègne, BP 20529, 60205 Compiègne CEDEX, France, dominique.lenne@utc.fr

Information and Communication Technologies have transformed the way people work and have a growing impact on long life learning. Organizational Learning is an increasingly important area of research that concerns the way organizations learn, and thus augment their competitive advantage, innovativeness, and effectiveness. Within the project MEMORAe2.0, we are interested in the learning centering subject. We developed an environment based on Topic Maps which

enables to navigate in a concept map. Therefore, users can access resources indexed by cartography concepts. In our approach, we chose to put forward the notion of course and not the structure. In this paper, we define the concept of Learning Organizational Memory, and we present Topic Maps and how they can be used to model a memory. Then, we present our approach to model our memory thanks to Topic Maps. Finally, we present briefly our environment.