Leipziger Beiträge zur Informatik: Band XXIV

# Information Wants to be a Topic Map

Sixth International Conference
on Topic Maps Research and Applications, TMRA 2010
Leipzig, Germany, September 29 – October 01, 2010
Revised Selected Papers

**Lutz Maicher**
**Lars Marius Garshol (Eds.)**

# Information Wants to be a Topic Map

Sixth International Conference
on Topic Maps Research and Applications, TMRA 2010
Leipzig, Germany, September 29 – October 01, 2010
Revised Selected Papers

## Volume Editors

**Dr. Lutz Maicher**

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

**Lars Marius Garshol**

Bouvet AS
Sandakerveien 24C D11
NO-0403 Oslo, Norway
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/2010

**Information Wants to be a Topic Map**
Sixth International Conference on Topic Maps Research and Applications,
TMRA 2010 Leipzig, Germany, September 29 – October 01, 2010
Revised Selected Papers
Lutz Maicher, Lars Marius Garshol (Eds.) – Leipzig, 2010

ISBN 978-3-941608-11-5

# Preface

The papers in this volume were presented at TMRA 2010, the International Conference on Topic Maps Research and Applications, held 30 September and 1 October 2010, in Leipzig, Germany. TMRA 2010 is the sixth conference in this annual series of international conferences dedicated to Topic Maps in science and industry.

The motto of the TMRA 2009 conference, "Linked Topic Maps", was about spinning a global web of interchangeable and linkable topic maps. Linked Topic Maps was about lightweight semantic integration and fusion of heterogeneous sources of structured and unstructured data, based on the integration model provided by Topic Maps. With the Topic Maps 2010 conference in Oslo the first applications of linked topic maps in industry emerged.

At the same time the stack of freely available Topic Maps software grew considerably within the last months. The technology is ready for the next level of scale. But scalability needs data. It was Kal Ahmed at the TMRA 2009 conference who coined the motto for this year's conference: "Information wants to be free. Information wants to be a topic map."

With the TMRA 2010 conference various approaches to make Topic Maps based data available was explored. There is a wide range of means for producing these maps: starting with generic Topic Maps editors, over web portals and domain specific desktop applications, to advanced approaches of exposing data sources as virtual topic maps.

Consequently new data sources generate new applications. Which new kind of applications are possible and already implemented? Do these applications require new data sources which are not available in the Topic Maps world today? Generally, all kinds of applications are of interest – the scaling, web-based ones and the applications on the enterprise level. They all have one thing in common: they all need information which behaves like topic maps.

The goal of the TMRA 2010 conference was bringing together all the ideas, concepts and implementations which will help to make information accessible and usable in a Topic Mappish way. The paradigm of the TMRA 2010 conference was that everything which behaves like a topic map is a topic map, irrespective of its origin. It's all about paving the way for global knowledge federation.

Stimulated by the success of the previous conferences the concept of TMRA was retained nearly unchanged. The conference was preceded by tutorials@TMRA 2010, a full day of in-depth tutorials. The main conference schedule was separated into two parallel tracks, providing a rich program for all interests. The Open Space sessions, once more smoothly moderated by Lars Marius Garshol, provided a light and exciting look at work which will most likely be presented at future conferences.

The "Topic Maps Elevator Pitch Contest at TMRA 2010" was a new feature of this year's TMRA conference. The participants got the stage for 60 seconds. This was their opportunity to explain the idea and the benefits of Topic Maps to the audience. To win the contest the crowd at TMRA had to be convinced with the pitch.

The TMRA 2010 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 36 members. Out of 39 submissions, 20 were accepted as papers for these proceedings.

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 and Benjamin Bock, for their tireless commitment to make TMRA 2010 a success. TMRA was organized by the Zentrum für Informations-, Wissens- und Dienstleistungsmanagement at the University of Leipzig. We thank Anika Jahn for her engagement in the organization of the conference and Peter Scholz for producing these proceedings. Furthermore we acknowledge the generous support by all our 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 2011.

Leipzig and Oslo,
September 2010

Lutz Maicher
Lars Marius Garshol

# Organization

TMRA 2010 was organized by the Zentrum für Informations-, Wissens- und Dienstleistungsmanagement e.V. (ZIWD) in Leipzig, Germany.

### Chairs of the Program Committee

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

### Program Committee

Marie-Hélène Abel, Université de Technologie de Compiègne, France
Kal Ahmed, NetworkedPlanet, UK
Frédéric Andrès, NII, Japan
Michel Biezunski, Coolheads Consulting, USA
Arnim Bleier, Topic Maps Lab at the University of Leipzig, Germany
Benjamin Bock, Topic Maps Lab at the University of Leipzig, Germany
Robert Cerny, Researcher, Germany
Darina Dicheva, Winston-Salem State University, USA
Ralf Eilbracht, Nexxor, Germany
Lars Marius Garshol, Bouvet, Norway
Lars Heuer, Researcher, Germany
Gerhard Heyer, University of Leipzig, Germany
Tobias Hofmann, Researcher, Germany
Stefan Kesberg, Nexxor, Germany
Aki Kivelä, Grips Studios Interactive, Finland
Tomáš Kliegr, Ptague University of Economics, Czech Republic
Giovani Rubert Librelotto, Universidade Federal de Santa Maria, Brasil
Stefan Lischke, Researcher, Germany
Lutz Maicher, Topic Maps Lab at the University of Leipzig, Germany
James David Mason, Y-12 National Security Complex, USA
Thomas Neidhart, Space Application Services, Belgium
Steven R. Newcomb, Coolhead Consulting, USA
Jan Nowitzky, Deutsche Börse, Germany
Sam Oh, Sungkyunkwan University, Korea
Jack Park, Researcher, USA
Rani Pinchuk, Space Application Services, Belgium
Jan Schreiber, Ravn, Norway
Alexander Sigel, Researcher, Germany
Michael Sperberg-McQueen, Black Mesa Technologies, USA
Kevin Trainor, University of Illinois at Urbana-Champaign, USA
Johannes Schmidt, Researcher, Germany
Stefan Smolnik, European Business School, Germany
Thomas Schwotzer, HTW Berlin, Germany

Volker Stümpflen, Helmholtz Zentrum München, Germany
Markus Ueberall, Researcher, Germany
Gerhard Weber, Nexxor, Germany

**Organization Committee**

Lutz Maicher, Topic Maps Lab, University of Leipzig, DE (Chair)
Benjamin Bock, Topic Maps Lab, University of Leipzig, DE (Chair)
Anika Jahn, Topic Maps Lab, University of Leipzig, DE (Chair)

**Sponsoring Organizations**

Networked Planet, Oxford, UK
Ravn Webveveriet AS, Oslo, NO
Norwegian Computer Society, Oslo, NO
Wandora Team, Helsinki, FI
Morpheus Kenntnistechnologie BV, Utrecht, NL
Mediafoundation of the Sparkasse Leipzig, DE
Topic Maps Lab, Leipzig, DE

# Table of Contents

## V Investigation Ontology Structure



## VI Semantic Integration



## VII Theoretical Topic Maps Research



## VIII Topic Maps on the Web

## IX Topic Maps in the Industry



## X Report from the Sessions

# Part I

# Topic Maps Frontends

# AToM$^2$ – a "web database" with Topic Maps Roots

Pavel Gardavský

AION CS, Ltd.
pavel.gardavsky@aion.cz
http://www.aion.cz/En/Default.aspx

AToM$^2$ is an acronym of "**A**ion **To**pic **M**aps engine **2**nd generation". AToM$^2$ is

- an application framework for building semantically oriented projects (like encyclopedias, legal systems, vocabularies, knowledge bases, sophisticated CMSs ...)
- a high performance and usability oriented feature-rich web database
- strongly influenced by Topic Maps concepts and slightly inspired by other semantic techniques and approaches

Purpose of this document is providing basic information on AToM$^2$ for the TMRA committee to decide if it is worth for presentation on TMRA 2010

## 1 AToM$^2$ raison d'être

Main activities of our company AION CS, Ltd. are focused on building legal information system, encyclopedias equipped with complex content management systems and presented particularly via web applications. These projects are mostly built on taxonomies, thesauri, associations, categorization into classes. Our problem was that every project needed its specific semantics and we needed some unified solution. In 2006–2008 we tested existing solutions (OKS, TMCore, TMAPI.Net, Wandora ...). As a result we decided (despite Lars's warnings) in early 2008 to develop our own Topic Maps engine – AToM. Structured reasons for this decision:

- every above mentioned solution had some bottleneck which made usage in our projects in some way difficult
- on one hand we didn't need all Topic Maps complexity and universality and on the other hand we needed some features not directly included in Topic Maps standards but demanded by our customers and projects
- because of plans in which AToM solution should be a core component of our future projects we need complete supervision on the source code and independency on somebody else
- and last but not least we need performance (responses preferably in $\mu$s ) and possibilities and scalability of new Microsoft technologies (.NET Framework 3, SQL Server 2008, TFS Team development etc.)

## 2 AToM$^2$ history

| | | |
|---|---|---|
| 5–11/2008 | AToM$^1$ | first implementation of the engine:<br>– broadening fulltext queries with context<br>– user query is e.g. “fruit”, results don’t contain “fruit” but do contain apples, bananas, oranges…<br>– AToM is an interface layer between query form and fulltext engine (Verity or Lucene), queries are broadened by a Topic Map<br>– XTM imports, management of the layer, SQL server storage<br>– example 1: the very beginning – experiments with Eurovoc converted in Topic Map, http://eurovoc.test.aion.cz (Czech only)<br>– example 2: several famous topic maps (incl. Opera) in several interfaces (ASP.NET, Ajax, Silverlight), http://atom.aion.cz |
| 2/2009 | AToM$^2$ | start of development of the new (complete in-house) engine |
| 4/2009 | AToM$^2$ preview | first bug-loaded public preview on following principles<br>– strict of the ontology layer and data layer<br>– WPF based Ontology designer, ASP.NET Instance editor<br>– 1 ontology = 1 AToM = 1 database = 1 URL<br>– ontology can be divided into separate topic maps ($\sim$ spaces)<br>– topic maps can either exist separately or can interfere |
| 7/2009 | AToM$^2$ alpha 1 | tons of bugs solved; first careful discussions about implementing in real projects; occurrences “reloaded” and renamed into properties<br>– enumerative 1/n, n/n properties “select”<br>– PSIs re-implemented and divided into “codes” and “identifiers”<br>– XHTML content property “text” (single cardinality) and “note” (multi cardinality) with comfort customizable editor<br>– another new properties “GPS” and “file” added scope massiveness, implemented as “folder” feature into ontology designer and used in instance editor for user interface customization |
| 10/2009 | AToM$^2$ alpha 2 | hundreds of kilograms another bugs solved; another property equipment enhancement<br>– names divided into “only one” name and “multiple” aliases<br>– new property “image” with complete management<br>– new powerful hierarchical property “group tree” – a multilevel taxonomy with choice of 3 types of element in every branch – 1/n, n/n, organizational element with robust database support; vector icon support for any class, association, role, property; first project: collaboratively developed Chinese Czech dictionary |

| | | |
|---|---|---|
| 12/2009 | AToM$^2$ alpha 3 | tens of kilograms of new and old bugs solved; preview of new end-user interface in Silverlight 3<br>– robust cascade search filters, prepare for faceted search<br>– various result list (grid, tiles, slider); launch of another projects<br>– picture database for a well known publishing house: categorization along various facet axes, picture relations to sources, licenses, usages in books, articles, editors based on "legacy association" principles, plenty of thousands of pictures imported<br>– launch of 3 year project: knowledge portal on NEC (Network Enabled Capability) problematic for University of Defense |
| 1–3/2010 | AToM$^{2.1}$ beta 1 | complete rewrite<br>– communication layer converted from direct SQL accesses to very fast implementation of WCF<br>– instance editor – eliminating of all 3rd party components and putting all interfaces on AJAX techniques; added new feature package necessarily needed in our document oriented projects: sort and strength of the association; development of new Silverlight framework for handling all features of property equipments |
| 4/2010 | AToM$^{2.1}$ beta 1 | launch of new project: European Legislation (EurLex), Czech and Slovak Collection of Law – some 400 thousand documents interconnected with millions of associations are the base for end-efficient knowledge portal |
| 5/2010 | AToM$^{2.1}$ beta 1 | adaptation to.NET Framework 4, Silverlight 4; estimate that we managed some 30–40% of our way to target |

## 3 AToM$^2$ Technology Background

– schemas, data, and most of the application logic stored in MS SQL Server 2008 structures
– IIS 7 used as a web server
– all code is written in C# (Visual Studio 2010); ASP.NET, AJAX used, but no 3rd party components because of full performance control necessarity
– project development is handled by Team Foundation studio 2010
– there is only one code for all AToMs, customization is made via parameters in web.config and of course project specific web application

## 4 AToM$^2$ (beta 1) Architecture

Following block scheme describes current architecture

### 4.1 Ontology designer

– complete ontology management, consists of 3 editors

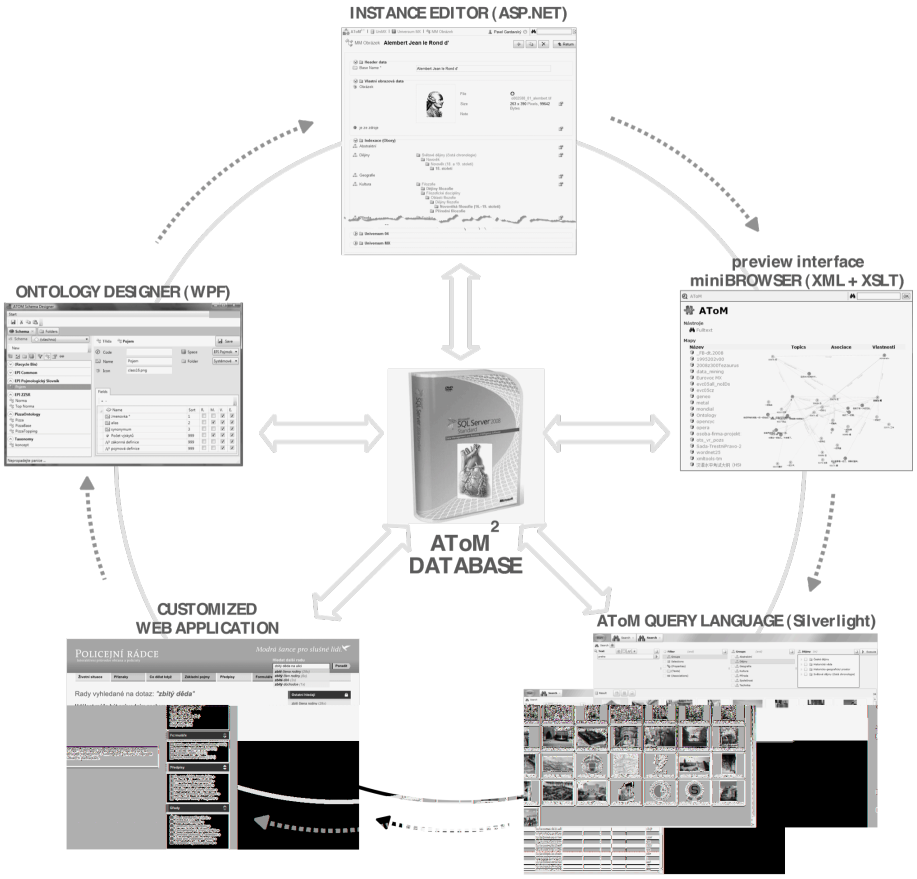



**Fig. 1.** AToM$^2$ beta 1 block scheme

- Space editor – managment of spaces (topic maps)
- Folder editor – managment of folders (scopes)
- Schema editor – management of classes (topic types), roles/associations and all the properties (occurrences) staff: names, aliases, variants (boolean, string, date, integer, decimal, float, string date & time, control which properties are indexed into fulltext index and which are not

– coded in WPF, nested in IE browser via XBAB technology
– future plans: complete rewrite in Silverlight,
– some screen shots examples (Figs. 2, 3)

### 4.2 Instance editor

– complete equipment for instances management

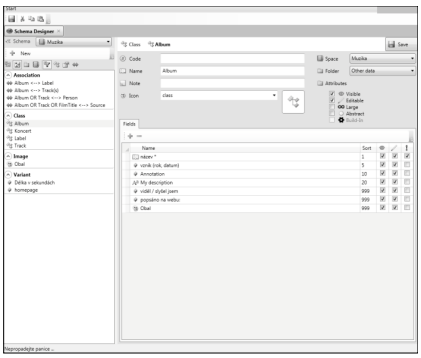


**Fig. 2.** Schema editor – Topic Type (Class) definition

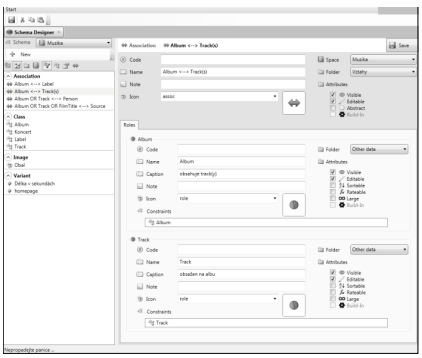


**Fig. 3.** Schema editor – Association management

- instance create, destroy, hide, duplicate
- personalized property and association editing

– written in ASP.NET, very optimized data communication in WCF
– future plans: complete personalized CMS, rewrite into Silverlight and many, many others
– some screen shot examples (Figs. 4–7)

### 4.3 Mini Browser

– review and preview of completed work, but also a very simple web site
– some screenshots examples (Figs. 8, 9)

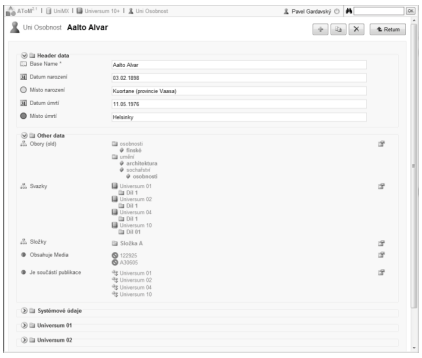


**Fig. 4.** Instance editor – Instance detail with group tree

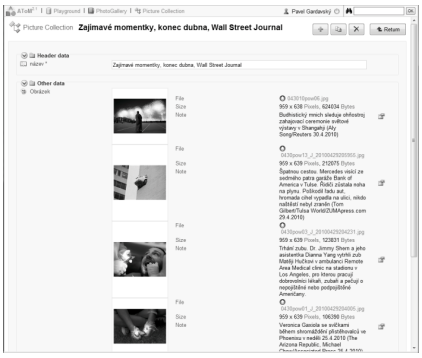


**Fig. 5.** Instance editor – Instance detail with images

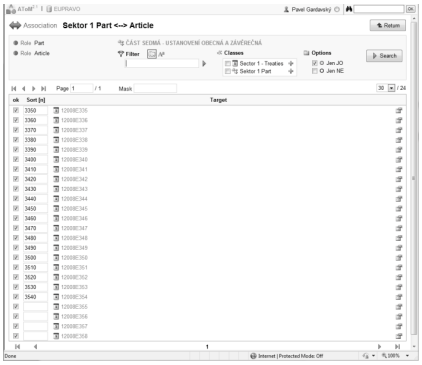


**Fig. 6.** Instance editor – sorted Association creation

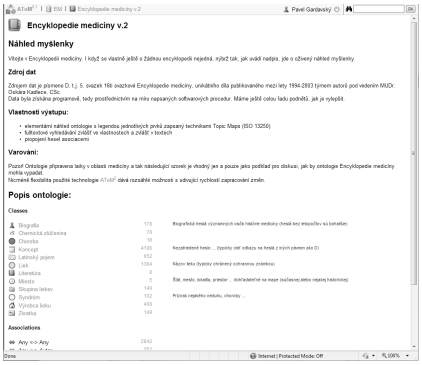


**Fig. 7.** Instance editor – space (topic map) entry page

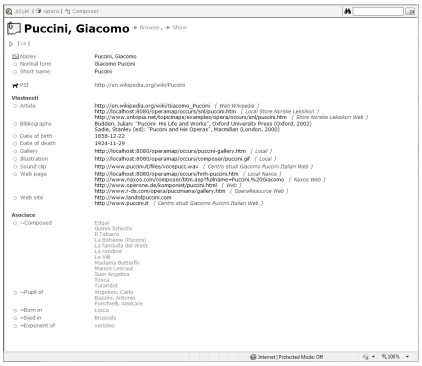


**Fig. 8.** Mini Browser – preview of instance

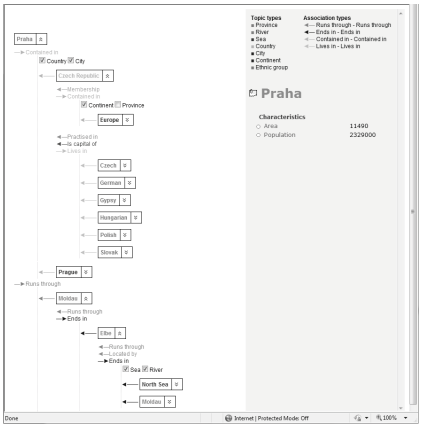


**Fig. 9.** Preview of association equipment

## 5 AToM$^2$ Live Examples

There is plenty of real examples which can be presented, some of them are mentioned in history table above. It is (almost) sure that until TMRA 2010 there will be another bunch of them

## 6 Some AToM$^2$ Future/Further Plans

- AQL – AToM Query Language with graphical GUI
- Association properties in the same way as Classes
- User/Roles/Application/Permission management
- Compatibility table with TMDM and TMCL
- Web application generator
- AToM$^3$

# Putting Topic Maps to REST*

David Damen and Maria Patriksson

Space Application Services, Leuvensesteenweg 325
B-1932 Zaventem, Belgium
\{david.damen,maria.patriksson\}@spaceapplications.com
http://www.spaceapplications.com/

**Abstract.** We introduce TROPICS, a specification for a topic map engine that fully conforms to the constraints of REST. TROPICS meets the needs of several use cases for topic maps identified in the ULISSE project, where topic maps are used to describe space experiments from various scientific disciplines. The design supports different granularity levels in topic map interaction, and offers solutions for resources in terms of versioning, navigation, querying and representation. We describe the resources provided by TROPICS to access topic map constructs, and touch upon the available media types that can be used to describe these.

## 1 Introduction

In the ULISSE[1] project topic maps are used to describe space experiments from various scientific disciplines [1]. In the course of the project we have identified several use cases for topic map interactions over the web. The human-centric knowledge representation of topic maps and the merging mechanism between topic maps contain an implicit promise of facilitated information discovery and sharing, a promise that can only be realised if topic maps are made available to the public through services with well-defined and uniform interfaces. TROPICS offers a solution to this problem.

Not surprisingly, the need for engine support of different granularity levels in this interaction has arisen, ranging from interaction specific to one topic map, to multiple topic maps, to all topic maps stored in the engine. The identified use cases, together with the identified needs for different information granularity, form the basis of the design of TROPICS.

## 2 TROPICS

### 2.1 REST and Constraint Compliance

The REST architecture describes constraints that emphasize scalability, generic interfaces and independent deployments to reduce interaction latency, enforce security and encapsulate legacy systems [2]. Conforming to these constraints is referred to as being RESTful .

* This work has received funding from the European Commission through the Seventh Framework Programme (FP7/2007-2013) under the Space Theme, under grant agreement Nr. 218815 within the ULISSE project.

[1] USOCs KnowLedge Integration and Dissemination for Space Science and Exploration

The central feature that distinguishes the REST architectural style from other network-based styles is its emphasis on a uniform interface between components, which is obtained through compliance with the four interface constraints: *Identification of Resources*, *Manipulation of Resources through Representations*, *Self-descriptive Messages* and *Hypermedia as the Engine of Application State*.

TROPICS complies with the interface constraints of REST.

**Identification of resources** REST uses resource identifiers to identify particular resources involved in an interaction between components [2]. On the web, resources that can be interacted with are typically HTML documents, but also images, video, etc. URLs are used to identify them.

The resources a TROPICS engine makes available are topic maps, topics and associations. IRIs[2] are used to identify topic map constructs. A TROPICS engine generates valid IRIs for topics and associations and stores them as item identifiers thereby conforming to this constraint.

**Manipulation of Resources through Representations** REST components perform actions on a resource by using a representation to capture the current or intended state of that resource and transferring that representation between components [2]. On the web MIME-typed documents serve as representations for resources. Depending on the MIME-type, a client requesting a representation of a resource can act on the result.

A TROPICS engine responds to calls with topic maps. Such a topic map can be represented using any of the standardized topic map formats, such as XTM 2.0, or de facto standards, such as JSON TOPIC MAPS (JTM) [3]. Similar to the web, TROPICS uses HTTP and allows the use of the same set of standard functions to interact with the resources it makes available. This means that a client can choose the format of a representation and that each format must have an associated MIME-type.

**Self-descriptive messages** REST enables intermediate processing by constraining messages to be self-descriptive: interaction is stateless between requests, standard methods and media types are used to indicate semantics and exchange information, and responses explicitly indicate cache-ability [2]. HTTP describes a number of standard functions[3] with well-defined semantics that can be used to interact with resources.

A TROPICS engine uses HTTP and allows the use of the same set of standard functions as those found on the web to interact with the resources it makes available. Care should be taken when implementing a TROPICS engine so it does not break the semantics of the standard HTTP functions.

**Hypermedia as the Engine of Application State**[4] is arguably the most difficult to understand and hardest to design concept of REST. For a resource to be compliant with this constraint, it must contain the links that are valid state transitions to other

[2] Internationalized Resource Identifiers

[3] GET, PUT, POST, DELETE and a few more

[4] HATEOAS

representations. A client should be able to start using a RESTful application by retrieving the representation of a single known URI (a bookmark). In this representation the client should be able to find links to other valid representations which in turn contain valid links to other resources and so on. On the web, hyperlinks and forms allow clients to transition from representation to representation. The information on how to use hyperlinks and forms is considered information that a client can know in advance, e.g. that form element contains a `method` attribute indicating which HTTP method should be used to submit the form to the server.

In order to explain how TROPICS fulfills this constraint, we must first define what *application state* entails for a topic map client. Compared to a web server that serves data, layout information and source code for applications, a TROPICS engine focuses on serving a data driven representation of knowledge to its clients. This focuses the possibilities for the use of a topic map engine towards:

1. **Navigation through knowledge** A client navigates from topic to topic through associations, or moves from one topic map to another.
2. **Querying** A client queries a TROPICS engine to
   (a) Immediately find a piece of information
   (b) Find a point in the topic map from which navigating the knowledge captured in it should start.

From the latter two sub points we can generalize the *application state* of a client of a TROPICS engine as the topic that the client is looking at. Valid *state transitions* are then the different ways the client can navigate away from the current topic. From this, three coarse-grained navigation steps can be identified:

1. **Intra-Topic Map navigation** Navigation between topics using associations. This kind of navigation can be further divided into several specialized cases, such as from a topic to its types, to its instances, to its role types, etc. For instance, navigating from the *Sun* topic to the *Earth* topic by following the *Orbits* association.
2. **Inter-Topic Map navigation** Navigation from a topic in a topic map to the same topic in a different topic map. For instance, navigating from the topic *Sun* in an astronomy topic map to the *Sun* in a solar radiation topic map.
3. **Merged navigation** Navigation from a topic in a topic map to the same topic as it appears in the merged result of several topic maps. For instance, navigating from the *Sun* topic in an astronomy topic map to the *Sun* topic as it appears in the topic map merged together from all space-related topic maps.

Hence, if a TROPICS engine wants to fulfill the HATEOAS constraints, it needs to put links for navigating the topic map space in its resource representations.

### 2.2 Media Types

TROPICS describes the media type(s) used for representing resources and driving application state. Existing media types should be reused when possible. Defining which methods to use on which URIs of interest should be done within the scope of the processing rules of a media type [2].

### 2.3 Resource Representation

An engine complying to TROPICS represents its resources as topic maps. In order to fulfill the HATEOAS constraint, it needs a way to represent links in its representations. Logical places to include this kind of links are:

1. As item identifiers of existing Topics.
2. As a Topic Map fragment in the result.

Both approaches are valid and can be used in conjunction, thereby turning a topic map representation of a resource in TROPICS into a combination of two parts:

1. **Informational Topics** Topics that contain the actual informational data representing the resource that was requested. These topics contain links for further navigation as item identifiers.
2. **Navigational Topics** Topics that solely contain information on how to navigate further in the URI space of the TROPICS engine, including:
   (a) Other parts of the topic map space linked to the current topic.
   (b) Other functionality provided by the TROPICS engine, such as searching.

The following formats can be used as media types in TROPICS, a MIME-type is proposed for each of them:

1. XTM 2.0/2.1: application/xtm2, application/xtm2.1
2. CXTM: application/cxtm
3. JTM: application/jtm

Some other popular topic map formats cannot fully support a TROPICS engine as they have no support for item identifiers on associations as described in the topic map data model (TMDM):

1. XTM 1.0
2. CTM
3. LTM

Note also that a topic map representation may not always be the best choice. A query resource might benefit from a different representation that is quicker to parse.

### 2.4 URI Operations

This section describes how clients of a TROPICS engine uses the URIs in topic map representations.

**Navigational Topics** All topics that are part of the navigation ontology or are instances of the topic types in the navigation ontology are navigational topics. Navigational topics describe themselves how to operate on the links they represent through occurrences, similar to a form element in HTML . The navigation ontology contains three topic types:

1. **Link** Instances of `Link` contain the following occurrences:

   (a) uri: a URI that identifies a resource that can be navigated to.
   (b) method: the name of the method that can be used on the URI [5].
   (c) description: a description of what `uri` points to.
2. **Parameter** Instances of `Parameter` represent parameters that can be appended to the URI of `Links` they are associated with through the `supports` association. When a `Parameter` is mandatory, a unary association will be attached to it. They contain the following occurrences:
   (a) alias: one or more entries containing the parameter name (this is typically a long form and a short form, e.g tms-include and ti).
   (b) datatype: indicates how parameter values should be formatted.
   (c) description: a description of what the parameter represents.
3. **Input** Instances of `Input` represent valid inputs that can be sent when operating on a URI of `Links` they are associated with through the `accepts` association. When using HTTP , inputs to URI go into the request body. They contain the following occurrences:
   (a) format: the data format that is accepted by the URI . There can be more than one acceptable format.
   (b) description: a description of what the input represents.

The graphical structure of the navigation ontology can be seen in Fig. 1. An example of how this ontology can be used to create navigational topics can be found in Fig. 2, represented using the GRAPHICAL TOPIC MAPS NOTATION (GTM) [4].

The example shows two navigation links, one to create a new topic and one to query the TROPICS engine. Creating a new topic can be done with a POST to `/topics` and supplying the topic, represented in either XTM2.0 or JTM, as input. Querying the TROPICS engine can be done through a GET on `/search`. The query itself should be provided as a parameter using either QUERY-STRING or Q as the parameter name. Both links also support TMS-INCLUDE as a parameter to choose on which topic map(s) the action should be executed.

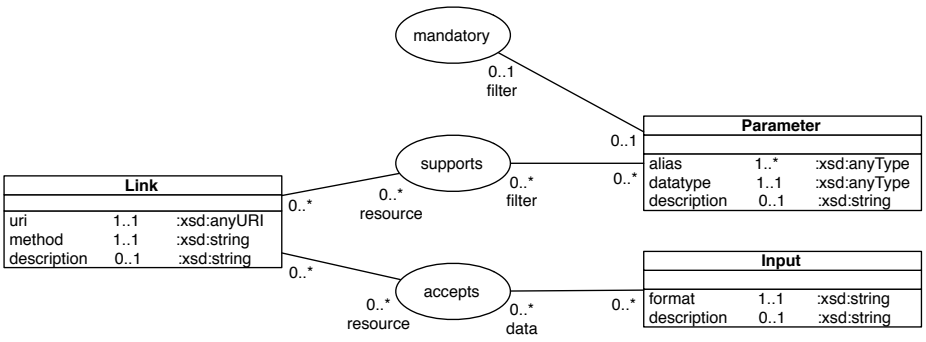



**Fig. 1.** Navigation Ontology for TROPICS

[5] When using HTTP , this is typically one of POST, GET, PUT or DELETE

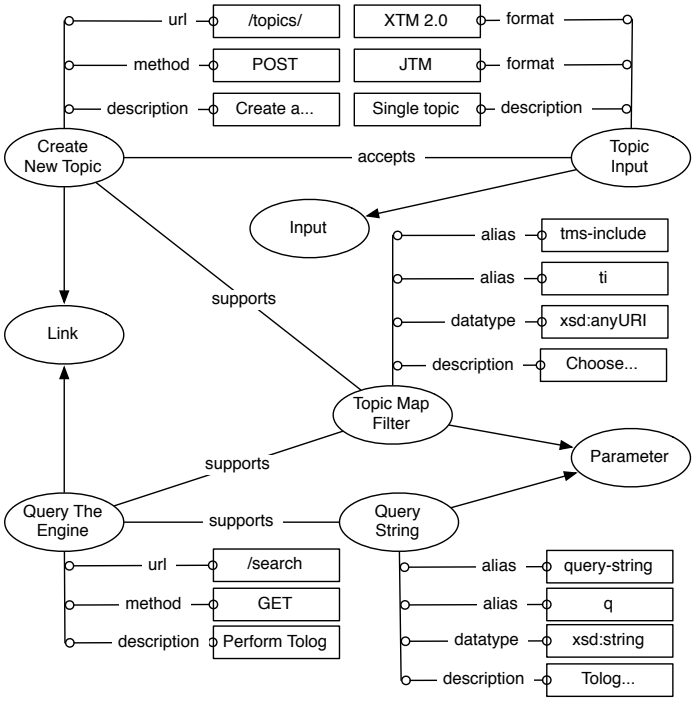



**Fig. 2.** Topic map fragment containing navigational topics

**Informational Topics** All topics that are not connected to topics in the navigation ontology are considered informational topics. A specialized information ontology is also introduced to represent the concept of topic map groups. This ontology contains two topic types:

1. **TopicMap** Each instance of `TopicMap` represent a topic map that is stored in the TROPICS engine.
2. **Group** Each instance of `Group` represent a topic map group created in the TROPICS engine. A `Group` can contain `TopicMap` and/or other `Group` instances.

The graphical structure of the information ontology is depicted in Fig. 3. Informational topics typically contain as an item identifier a URI that can be further acted on. In HTTP, POST creates a new resource, GET is used to retrieve a representation of the resource, PUT updates the resource and DELETE removes it.

### 2.5 Navigation

As previously mentioned, a core use case of topic maps is knowledge navigation. In this section we describe the resources provided by TROPICS to access topic map constructs.

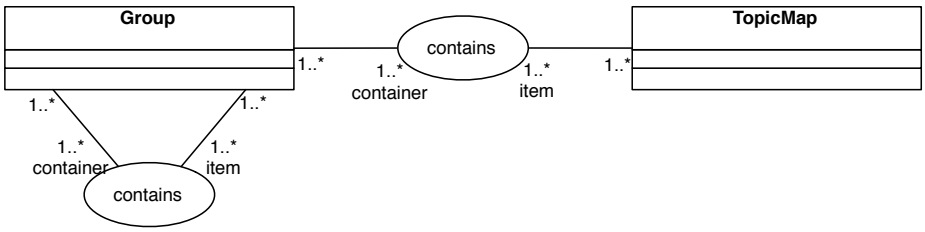



**Fig. 3.** Information ontology for a TROPICS engine

**Topic Maps** Resources representing complete topic maps.

`/topicmaps/`: A collection resource representing all topic maps stored in the TROPICS engine. It can also be used to create new topic maps in the TROPICS engine, i.e. by using POST.

`/topicmaps/{id}`: A resource representing a complete topic map as stored by the TROPICS engine.

**Topic Map Groups** Resources representing a collection of topic maps. A topic map group can contain zero or more topic maps and zero or more other topic map groups.

`/groups/`: A collection resource representing all topic map groups stored in the TROPICS engine. It can also be used to create new topic map groups in the TROPICS engine.

`/groups/{id}`: A resource representing a single complete topic map group as stored by the engine.

**Topics and Associations** Resources representing individual topics and associations. Parameters on the URI of the resource distinguish whether a topic or association should be interpreted as it occurs in a single topic map, multiple topic maps or all topic maps.

`/topics/`: A collection resource representing all topics stored in the TROPICS engine. It can also be used to create new topics in the TROPICS engine.

`/topics/{id}`: A resource representing a single topic as stored by the engine.

`/associations/`: A collection resource representing all associations stored in the TROPICS engine. It can be also be used to create new associations in the engine.

`/associations/{id}`: A resource representing a single association as stored by the engine.

### 2.6 Querying

Next to navigating through the topic maps, another core use case is searching for information. This functionality is supported by a dedicated resource.

`/search`: A resource that takes the query string as its parameter. Which query language that is to be used is defined through a parameter. The availability of a specific query language depends on the implementation of the engine.

**Table 1.** Resource versioning suffixes

| Suffix | Notes |
|---|---|
| `/versions/` | A collection Resource that returns all versions of the Resource it is attached to. |
| `/versions/{x}` | The `x` indicates a version number. It can be used to retrieve a specific version of the Resource. |
| `/versions/latest` | Returns the latest version of the Resource. Not using a versioning suffix on a versioned Resource redirects to `/version/latest`. |

### 2.7 Versioning

When talking about versioning of RESTful applications, it is important to make a distinction between versioning resources described in the API and versioning the application itself.

**Resource Versioning** When different versions of a specific resource can be retrieved, then a set of common versioning suffixes can be used to access those different versions. Table 1 depicts these suffixes and their meanings.

In case a resource in TROPICS is versioned using this approach, this is indicated in the resource description. Only a HTTP GET can be called on a resource with a versioning suffix.

**Application Versioning** With regards to versioning the application itself, it is expected that the root of the URI space of a specific API contains a path fragment identifying the version of the API, for example http://example.org/api/v1/.

In this URI , the appended `v1` indicates the version of this specific API. Having a version identifier allows running multiple versions of the same API on one server and makes it easier to migrate from one version to another.

## 3 Related Work

Several other remote access protocols have been described and developed for accessing topic map content from a topic map engine.

TMRAP is an abstract web service interface for remote access to topic maps [4]. It has been implemented in Ontopia[6] and both HTTP and SOAP bindings exist.

Topincs is a RESTful web service interface for retrieval and manipulation of topic maps [3]. Topincs has been defined and implemented by Robert Cerny in the Topincs Wiki. Development of Topincs also brought forth the JTM.

TMIP is a topic map aware protocol following the REST architecture which built heavily on a previous version of the Topic Maps Query Language [6].

[6] Ontopia is a Java-based, open source topic maps engine, http://www.ontopia.net/

## 4 Concluding Remarks and Future Work

We have introduced TROPICS and shown how to bring REST and Topic Maps together, without breaking the inherent constraints of REST.

We propose the use of existing Topic Maps representation formats, such as XTM and JTM, as media types to represent TROPICS resources. While this removes the need to introduce yet another data representation format, the performance aspects of this choice remain unchecked. Especially for resource representations containing a large number of navigational links, the overhead could grow quite large.

An engine in compliance with TROPICS will be implemented during the course of the ULISSE project. The engine will be implemented as an open source project in Ontopia.

## References

1. Damen D., Pinchuk R., Fontaine B.: Creating Topic Maps Ontologies for Space Experiments. In: Maicher L.; Garshol, L.M.: Linked Topic Maps, Proceedings of the Fifth International Conference on Topic Maps Research and Applications (TMRA'09)
2. Fielding R.: Architectural Styles and the Design of Network-based Software Architectures, Doctoral Dissertation, University of California, Irvine, 2000
3. Cerny, R.: Topincs – A REST*ful* Web Service Interface For Topic Maps. In: Maicher L.; Sigl, A.; Garshol, L.M.: Proceedings of the Second International Conference on Topic Maps Research and Applications (TMRA'06), Leipzig; Springer LNAI 4438, (2007)
4. Thomas, H.; Redmann, T.; Pressler, M.; Markscheffel, B.: GTMalpha ñ Towards a Graphical Notation for Topic Maps, 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
5. Garshol, L.M.: TMRAP – Topic Map Remote Access Protocol. In: Charting the Topic Maps Research and Applications Landscape, Heidelberg, 2006
6. Barta, R.: TMIP, A RESTful Topic Maps Interaction Protocol. Extreme Markup 2005, Montreal, Canada. http://conferences.idealliance.org/extreme/html/2005/Barta01/EML2005Barta01.html

# Designing a GUI Description Language with Topic Maps

Lukas Georgieff

Worms University of Applied Sciences
Erenburgerstr. 1967549 Worms, Germany
lukas.georgieff@hotmail.com

**Abstract.** Topic Maps (TM) is a powerful standard family to describe real world scenarios in computer processable data structures. On one hand it is so generic that arbitrary ontologies can be defined on the other hand this limitless concept brings up some problems when visualising this information to users who are not familiar with Topic Maps. To make TM databases applicable for end users it is necessary to solve the visualising problems of such a generic concept without enforcing limitations to it. This proposal presents the concepts of a description language to be created to design a graphical user interface (GUI) for specific ontologies defined in Topic Maps.

## 1 Introduction

The strength of Topic Maps [1] is the opportunity to define arbitrary ontologies and to define its structure using a standardized Topic Maps Constraint Language (TMCL) [5]. However this brings with it the problem for TM systems when displaying its information, since the same system is able to manage any number of topic maps respectively ontologies. These ontologies have different scopes, different purposes and of course different structures.

There are several possibilities to circumvent the usage of one generic GUI for all working TMs in a system:

- Usage of a full generic GUI with respect to the TM standard, i.e. the screen is presenting more or less a graphical version of the XTM [2,3] or CTM [4]. It is a progress given to some syntactical pitfalls eliminated by the system. But the user still has to understand every step of his action.
- Use of a generic GUI with respect to the ontology, i.e. the GUI displays a mask corresponding to the defined ontology via the TMCL – but also this approach is rather for administrative than for end users.
- Separate GUIs for different ontologies, i.e. the system knows which GUI is defined for each ontology and decides on demand what GUI is delivered for a specific user request. This mechanism is user-friendly but the administrators have to develop and manage several GUIs in parallel and every defined user-interface is restricted to a single ontology.

A new approach based on the information structure instead of the TM standard itself is necessary to enable a generic GUI. Such an approach should allow the definition of

a GUI not only for one particular but for all ontologies without limiting the Topic Maps standard itself.

After developing and using Isidorus-UI [6,7] it turned out that one improvement is inevitable – the improvement of the applicability for end users.

## 2 Isidorus-UI

Isidorus-UI [6,7] was a first approach to generate GUIs depending on the defined ontology by examining the TMCL constructs.

The created screen looks almost like the XTM 2.0 standard with similar descriptions and fields. Therefore we must assume that the user knows the basics of Topic Maps, e.g. the meaning of topics, occurrences, associations, roles, etc.. Instead of reaching the end users we just provide an administrative interface for TM insiders what limits the targeted audience.

Nevertheless, Isidorus-UI simplifies the TM administration by generating GUI from TMCL. The administrator interacts with a mask that is XTM close but also adapted to the ontology, what means

- all necessary objects are displayed (all identifiers, names, occurrences, associations, roles,...) in full view
- the operator only needs to fill in the necessary data or choose the correct (sub-) types for names, occurrences, roles, ...
- the entire input data is validated by the UI corresponding to the defined TMCL. Especially the generated fields and the final validation eliminate a lot of pitfalls when editing topic maps.

### 2.1 Restrictions

Generating a GUI from TMCL assumes that the user has knowledge about the Topic Maps standard. Additionally there are some restrictions to the GUI itself.

Since Isidorus-UI maps the defined TMCL to a screen, it is necessary to understand the effect of the Topic Maps Constraint Language.

Information belonging to a particular ontology is distinguished by the information and by the structure it provides. Ported to the TM standard it means that an ontology consists of certain constructs of predefined types and content within these constructs. Therefore all restrictions defined by these constraints effect the TM constructs themselves, i.e. representing an ontology by examining the TMCL leads to the fact that the UI must represent the defined TM constructs in a graphical manner.

It is the principle of Isidorus-UI to display one specific requested topic and all associations in which the topic is a role player. Creating or modifying several associated topics means to request every topic in particular and editing itself or it's associations. This process is very time consuming and the meaning of topics, associations and roles must be clear to the user. If several sub-classes of a role type are defined, the situation becomes even more complicated for people not familiar with TM.

Figure 1 demonstrates a generated GUI for a straightforward TMCL-definition with very few defined types and constructs. But even in this example the amount of generated fields can act as deterrent for non-TM users.

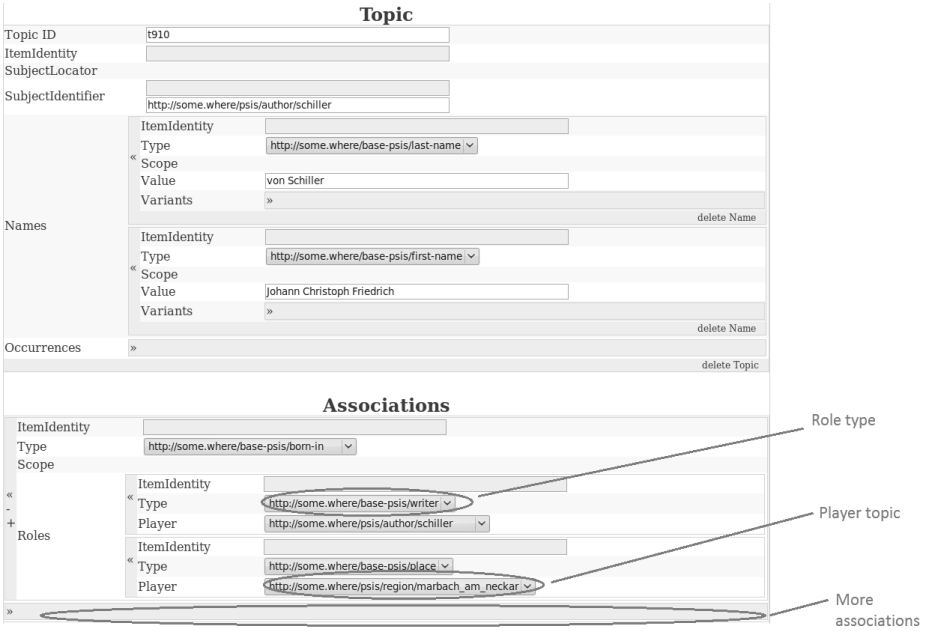



**Fig. 1.** Isidorus-UI

## 3 Designing a GUI Description Language (DGL)

The GUI generated by Isidorus shows one important approach – the flexibility to be used for every defined ontology. But for end users there is still a barrier of the TM internals. A further approach must be designed to generate GUIs for all possible ontologies decoupled from the TM internals and therefore user-friendly.

### 3.1 Goals

At the TMRA in 2009 Khalil Ahmed said "*Information wants to be free. Information wants to be a topic map*", two outstanding sentences during the conference in 2009. The TMRA's motto for this year is the second sentence "*Information wants to be a topic map*" the first part is an implicit assumption.

But what subsumes "*information wants to be free*"? - Not only the availability of information for everyone, but also the usability of this information for everyone meaning there should be a bridge between pure TM engines and practical-oriented end-users.

Our goal is to initiate a process of making TM more user-friendly by generating GUIs decoupled of TM internals for a particular ontology.

It is necessary to define an intermediate layer that allows the experts a generic way to define a user-friendly GUI adapted to a particular ontology, so that information is easy-to-get, easy-to-use, easy-to-extend and of course easy-to-administrate.

### 3.2 Starting Point

The original idea for a GUI Description Language (GDL) came up while using Isidorus-UI and defining a new ontology in the context of the TextGrid [16] project. Since the system should be administrated by non-TM-users the ontology was limited regarding the TM-capabilities. Some constructs were defined with the focus on Isidorus and the usability of the GUI, therefore the entire TM standard wasn't used, i.e. fewer sub-classes, fewer associations, ...

One starting point might be to extend TMCL:

- to define a possibility to provide default-values. This would simplify especially the entries which own a regular expression, e.g. the identifiers. So a given regular expression of the form "`^http://some.where/psis/base/.+`" provides the default-value "`http://some.where/psis/base/`" where the user only needs to enter the id-suffix.
- to define a possibility to hide fields that are static or predefined, e.g. item-identifiers or the topic-map-ids don't concern the end-user and can be hidden.
- to define the possibility to change the default layout of a construct.

This could be the first step to improve the usability for non-TM-users.

But creating and editing partial modeled data, i.e. topics related via associations still have to be processed in several steps, is very time intensive since the TMCL defines the overall relations as an ontology but is incapable of expressing the context in which the data is entered or manipulated.

Hence simply extending the TMCL is still not enough to improve the usability for non-TM-users. A new description language must be defined – an ontology to map TM data onto GUIs.

Not only must this language understand the constructs defined in TM, i.e. mainly topics, names, occurrences, associations and roles but also needs a mapping on concrete GUI widgets/components that can be visualized e.g. in a web browser or any other GUI implementation. The challenge is to combine these two requirements and find a simple and powerful language to map the abstract definition onto a real, graphical system.

### 3.3 Detailed Concept

Key point of the GDL is the opportunity to decide what information is in interest for the end-user and therefore should be presented to him at one glance on the screen. For that reason the particular topic- and association- types have to be chosen, linked and marked as one logic construct in order to be represented on the GUI.

Figure 1 shows the topic "Schiller", a German poet, and all its referenced associations. In contrast a GDL-driven GUI displays instead of the TM-associations the actual referenced information and, of course, the information placed in the topic "Schiller", i.e. a mask with the characteristics of a poet, like a name, date of birth but also a mask with all poems written by Schiller and maybe the place of birth. Hence the TM-associations are not needed anymore, it is not necessary for the user to understand the concept of associations and roles. Of course, all the sections displaying certain constructs must be mapped correctly back to TM, thus the GUI mask has to provide the vital information

about the construct itself, e.g. the type it is an instance of and the constraints defined by the TMCL.

But what to do with the remaining names, variants, occurrences, types and scopes?

For the purpose of simplicity, these elements should also be mapped to a corresponding mask. An example for an occurrence could be the poet's date of birth. In the TMCL driven client the topic frame appears with an occurrence frame containing all possible occurrences. The end user has to choose the correct (sub-) types, maybe one or several scopes, he also could set an item-identifier and after finishing these steps the user enters the actual data, the date of birth. All the steps prior to entering the date of birth can be achieved by the GDL. The item-identifier can be set automatically or even be hidden, the type and the scope can also be set automatically. If any user interaction is necessary, the GDL can define the corresponding mechanisms giving the user the chance to interact with several elements, e.g. by defining a possibility to set the scope, maybe a language, etc.

The last TM artifacts are the topic identifiers. They are very important for addressing constructs in TM but no end-user is happy to enter full URIs [8] for every single topic, especially not if there are such complex regular expressions as in the example of Fig. 1 "http://some.where/psis/author/schiller". These values can be generated automatically derived from other fields and are hidden from the end-user.

Essential for a GUI that can be widely accepted is also the naming of existing fields. Instead of naming them "Topic", "Name", "Occurrence", "Type" or "Scope" it is helpful to name them corresponding to their meaning, e.g. "Author", "First name", "Date of birth" or "Language". Reading these terms every user understands at first glance the intended meaning of the fields.

Acceptance can also be increased by an appealing GUI, i.e. the possibility of an individual defined mask should be offered; not only for a particular ontology but also from a graphic designer's point of view. This can be achieved e.g. by using CSS [9], especially for web clients, but any arbitrary language which can influence the style of GUIs can be used as well by e.g. embedding the code into specific occurrences.

Figure 2 shows an idea for a simplified mask design that could be generated for the same ontology shown as in Fig. 1. The mask looks very simple since all the TM complexity existing in Fig. 1 is hidden by the defined GDL.

This mask is focused on the creation of authors and poems. The user is able to enter directly the poet's characteristics and his poems.

The place of birth can be chosen from a drop-down-box, i.e. the user cannot create new topics of the type "place" directly. Such restrictions can be used to limit the accessibility of the user, but the GDL could also define a sub mask for this field if requested.

### 3.4 Interoperability with RDF

Besides interacting between the TM-backend and the user-frontend the GDL can also be used to interact between an RDF-backend [10] and a corresponding user-frontend.

Since the layout of the GUI is described as an ontology it is not limited to a particular technology. A corresponding ontology could also be defined by using RDFS [11] and OWL [12]. In the end the constructs will differ but with a sophisticated mechanism the

semantic meaning will remain and a mapping between the RDF GDLs and the TM GDLs is realisable as well.

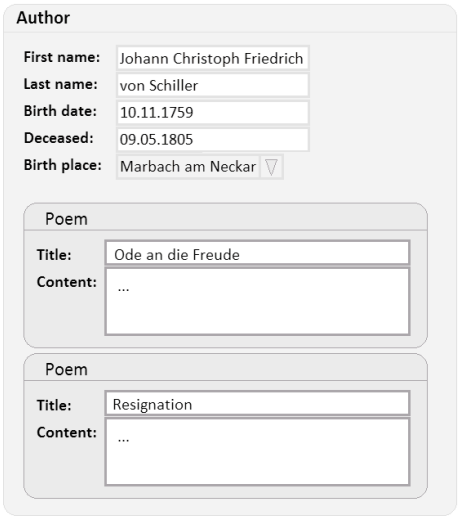



**Fig. 2.** Example of an ontology mask

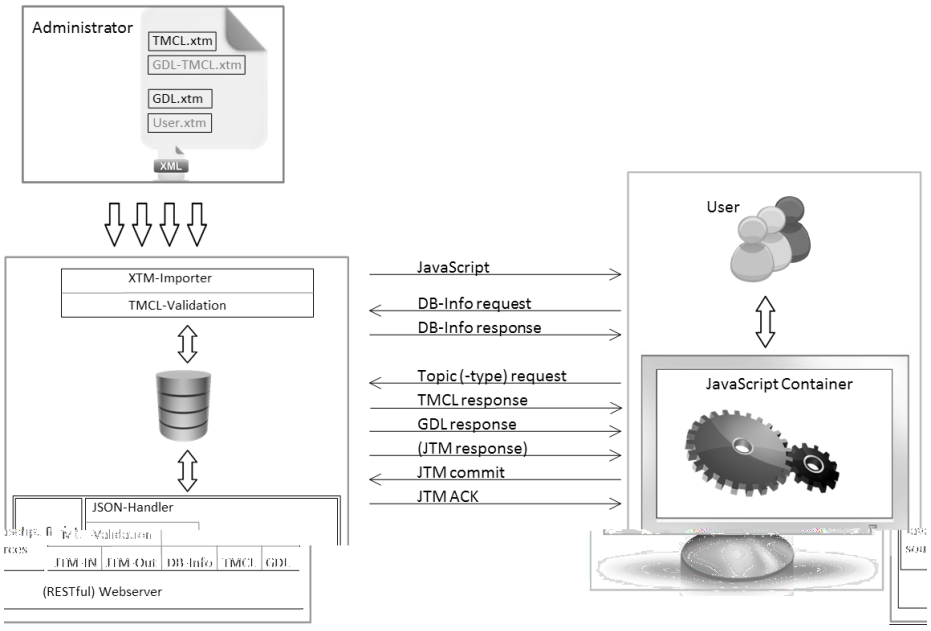



**Fig. 3.** Example of system providing GDL

### 3.5 An Exemplary Implementation

Figure 3 gives an overview about a hypothetical system using GDL to visualise the backend data to the user. It is based on Isidorus-UI. The mechanisms are very similar to the original system.

1. the TMCL files are imported, i.e. one TMCL for the user ontology and another for the GDL definition; this ontology is predefined by the system
2. the actual GDL content is imported
3. already existing user data can be imported optionally via the XTM-importer
4. the user requests the JavaScript files processed by the user's web browser
5. the client requests database information to create a general overview of topic (-types), etc.
6. the client requests a particular topic
7. the server answers with a JTM-fragment [17] including all necessary TM constructs and with the TMCL and the GDL information needed for this request
8. the client generates the corresponding GUI-mask
9. the user interacts with the generated GUI

## 4 Problems

The introduced concept comes with an additional layer used in a Topic Maps system. Since TM-processing can be very time intensive an additional step for every user request may cause extensive operations in the backend system and influence the performance in a negative manner. A simple solution is to save the generated GDL and TMCL data for a requested topic in a temporary object that is created once and only has to be serialised afterwards. There is no need to traverse the entire definition every time a topic (-type) is requested.

GDL files usually are TM-ontologies. This recursive way of expressing the GUI requires an additional system or tool to model the GDL-file itself. Editing XTM or CTM files is very time intensive and error-prone. Additional systems and tools that are able to generate ontologies are necessary to increase the production and usability of the introduced concept. A tool which allows the user to create a GUI via drag and drop and translates the created GUI into an XTM can also be a possible solution.

Nowadays many mechanisms supporting GUI programming, e.g. HTML [13], Java Swing [14],.Net [15] only to name a few, exist. All these systems have different models and different paradigms. The challenge is to define an ontology that abstracts most differences of the used GUI languages.

If the TMCL is not used as the last criterion to collect data, problems arises when the TMCL and the GDL differs for the same user-ontology. The question is how to place the GDL beside the TMCL. Must these ontology definitions interact or do they both stand as single instances for different purposes?

## 5 Conclusions

While implementing and using Isidorus-UI some weaknesses of a TMCL driven GUI were becoming clear. A TMCL generated user interface is restricted and visually too

tightly coupled to TM constructs, i.e. a lot of information that is not in the scope of interest of the end-user is generated and displayed.

With a definition of an intermediate language between the TM-backend and the GUI-frontend a lot of usability-problems can be solved. The gap between the TM-internals and the actual produced GUI can be closed or at least minimized. This abstract definition can be used on various systems and therefore is neither bound to a particular ontology nor to a particular GUI programming language. It is even not limited to Topic Maps.

GDL is a way to restrict the limitless of TM for the end-user's benefit. It simplifies the interaction with the TM-system and concurrently does not influence the strength of Topic Maps. If these goals are realisable for all practical purposes must be discussed in a future paper dealing with the implementation of the GDL.

## Acknowledgements

The author thanks Marc W. Kuester and Christoph Ludwig for thefruitful discussions of the ideas this paper is based on and for theirreviews of drafts of this article.

## References

1. ISO/IEC 13250 standard, available: http://www.isotopicmaps.org/TMRM/TMRM-7.0/tmrm7.pdf, [November, 2007]
2. XML TM 1.0 standard, available: http://www.topicmaps.org/xtm/, [August, 2001]
3. XML TM 2.0 standard, available: http://www.isotopicmaps.org/sam/sam-xtm/, [June, 2006]
4. TM Compact Syntax draft, available: http://www.isotopicmaps.org/ctm/ctm.html, [March, 2010]
5. TM Constraint Language draft, available: http://www.isotopicmaps.org/tmcl/2010-03-25/, [March, 2010]
6. Gießmann, L., Kuester, M.W., Ludwig, C.: "Isidorus-UI: Generating a User Interface with Topic Maps Constraint Language and JavaScript Object Notation", TMRA 5th International Conferences on Topic Maps Research and Applications, Leipzig, 2009
7. Project homepage, http://www.isodor.us, [July 2010]
8. RFC 3986, available: http://tools.ietf.org/html/rfc3986, [January, 2005]
9. W3C Recommendation, available, http://www.w3.org/TR/CSS1/, [December 1996]
10. W3C Recommendation, available: http://www.w3.org/TR/rdf-concepts/, [February, 2004]
11. W3C Recommendation, available: http://www.w3.org/TR/rdf-schema/, [February, 2004]
12. W3C Recommendation, available: http://www.w3.org/TR/owl-semantics/, [February, 2004]
13. W3C Recommendation, available: http://www.w3.org/TR/1999/REC-html401-19991224/, [December, 1999]
14. Project homepage, http://download.oracle.com/docs/cd/E17409_01/javase/tutorial/uiswing/, [July 2010]
15. Project homepage, http://www.microsoft.com/net/, [July, 2010]
16. Project homepage, http://www.textgrid.de
17. JSON Topic Maps notation, available: http://www.cerny-online.com/jtm/1.0/

# Part II

# Practical Topic Maps Research

# Modeling Units of Measurement

Xuân Baldau

xuan-measurement-units-2010-tmra.de@academia.baldauf.org

**Abstract.** Units of measurement are used everywhere where quantities are stored, albeit often only implicitly or even tacitly. Current topic maps are no exception. However, the nature of Topic Maps – to provide explicit *disclosure* in general – asks for explicit *disclosure* of the applicable unit of measurement of a given quantity in particular. The TMDM has no special support for units. This paper looks into how units can be used consistently within Topic Maps, and how a units topic map can be autogenerated using a Ruby DSL.

## 1 Introduction

"A volleyball should have a circumference of 65 to 67, and a mass of 260 to 280. Its acceleration towards earth is about 9.81. This means that the gravitational force on the ball is about 2.54 to 2.75."

No. Quantities without their proper units are like topics without subject identifiers, units provide the link between quantities and the real world. Thus, quantities everywhere (both in the sentences above and in Topic Maps in general) should have their units indicated.

Thus, units should be used in Topic Maps[1]. But what does "use" actually mean? When using units in a Topic Maps application, we may want to do, among other things, the following:

1. represent quantities and their units consistently in the TMDM [ISO13250-2]
2. display of units, and quantities with units, to humans (in the proper language)
3. convert quantities in one unit into quantities in another (compatible) unit
4. parse text into a quantity and a unit
5. refer to units conveniently (e.g. when authoring a topic map).

This, in turn, leads to following questions:

1. How can a particular quantity and its particular unit be properly represented using the TMDM?
2. How can a unit itself be represented in Topic Maps? What are the PSIs of units? What are their names? Is the unit *joule* the same unit as the unit *newton times metre*?

[1] Surprisingly, there has been little effort to properly and uniformly represent measurement units both in the Topic Maps and the RDF community. The most prominent approach for representing units in the RDF community is the Measurement Units Ontology [MUO2009], which itself builds on [UCUM2010]. Another (incompatible) approach in the RDF community is [Leal2002].

3. How can non-SI-units (such as *inch* or *U.S. liquid gallon*) be represented?
4. How can the relationships between units (e.g. one centimetre is the same as 1/91.44 yard) be represented?

We also have to consider common pitfalls, such as mixing up megabyte (=1 000 000 byte) with mebibyte (=1 048 576 byte) or simply prepending a prefix to a SI base unit (e.g. yielding "microkilogram")

The remainder of this paper is structured as follows: Section 2 deals with, given we know how to represent units and their interrelations in general, how to represent a particular quantity which may have a unit attached to it. Section 3 asks how a units ontology can look like given the use cases above. Section 4 summarizes further questions to be resolved.

## 2 Representing quantities with units

### 2.1 Unit of a quantity to be represented as special string, or as topic?

One of the first observations is that the precise data type of a quantity (e.g. integer, floating point, decimal, rational number, fuzzy number,...) is independent of the unit attached to the quantity. This means that if there is a datatype for each type of number (e.g. xsd:integer, xsd:decimal,... [XMLSchemaPart2]) (see Fig. 1), then a separate datatype supporting units would be in need to be invented for each type of number (e.g. xsd:integer_with_unit, xsd:decimal_with_unit,...). This style is infeasible, as it would manifest forever the boundary between quantities without units and quantities with units. Furthermore, this style would prompt the question for canonical identifiers for units (e.g. "m/s$^2$" or "m/s^2" or "m·s^-2" or "m^1·s^-2" or "metre per square second" or "meter per square second"). Topic Maps elegantly avoid the question of canonical identifiers by allowing multiple identifiers for a topic.

Thus, a particular unit, when used with a quantity, should be represented by a topic, not just a simple (sub)string.

### 2.2 Unit per quantity or unit per set of quantities?

It may be tempting to declare a unit for a particular quantity by annotating the occurrence type of the respective occurrence (see Fig. 4). Then, this declaration would be needed to be done only once, and only in a central place (e.g. a central topic map where all occurrence types for a particular application are defined). However, this centrality is too rigid. Consider two topic maps about people. One topic map measures heights in centimetre, one topic map measures heights in foot. If both topic maps are merged, then the occurrence type height would be measured in two different units. While this would be technically true, it would be no longer possible for the Topic Maps application to decide which quantity is actually measured in which unit. Thus, units need to be indicated for each quantity individually, not just once for all instances of a certain set of quantities.

There are some candidates of how to represent that Sarah has a height of 168cm (examples in CTM):

1. Use special data types:

```
Sarah
  isa person;
  height: "168 cm"^^xsd:decimal_with_unit.
```

2. Use scope (scope style):

```
Sarah
  isa person;
  height: 168 @unit:centimetre.
```

3. Annotate the occurrence (annotation style):

```
Sarah
  isa person;
  height: 168 ~ [is_measured_in(unit:centimetre)].
```

4. Annotate the occurrence type:

```
Sarah
  isa person;
  height: 168.

height
  is_measured_in(unit:centimetre).
```

**Fig. 1.** Different styles to represent quantities with units

### 2.3 Scope vs. annotation

Figure 2 and Fig. 3 differ slightly in their meaning. The former says “Sarah’s height is 168, in the context of centimetre.”, the latter says “Sarah’s height is 168, and this is measured in centimetre.”. It is difficult to decide between the two.

One the one hand, the scope-style is shorter than the annotation-style, in nearly any TMDM representation (be it CTM, XTM, or many internal formats of generic topic maps engines). Also, when looking at the sentences “Sarah’s height is 1.68, in the context of metre.” and “Sarah’s height is 5.51 in the context of foot.”, then the themes of a scope look like factors of when the statement is true. The and-semantics of scope themes in general ([Garshol2008Scope]) and the multiplicative semantics of scope in this application are certainly related (the common mapping of truth-values to 1 (true) as well as 0 (false) just needs to be completed by mapping “and” to multiplication). However, when going down this path further, then separate themes for prefixes would also be imaginable, like “height: 168 @prefix:centi,unit:metre” (if themes are factors,

then a factor may be factorized further). This in turn would also open the door for multiple prefix themes like “height: 168 @prefix:deca,prefix:milli,unit:metre” as well as “height: 168 @unit:metre,prefix:centi” (as themes of a scope are not ordered). Viewing units as contexts for quantitative statements still does not perfectly match the intention of units on the one hand and the intention of scope on the other hand. In short, the scope-style can be awkward.

When annotating occurrences (as in Fig. 3), then units are not part of the occurrence('s identity) itself, but they are additional information. Units being syntactically optional may be considered as a bug (as units can be considered to be essential to a quantitative statement), but this *non-intrusiveness* may also be considered as a feature, as it is a very common mistake to ignore disclosing the units of quantities, and in case of the annotation-style, this mistake can be fixed without changing the structure of the affected occurrence. (In case of the scope-style, the affected occurrence('s identity) would be changed by adding a unit to the scope of the occurrence, giving rise to further problems when merging with other topic maps.)

Note that the annotation-style also has a kind of irregularity. Consider 2 topic maps with following content:

```
Sarah
  isa person;
  height: 168 ~ [is_measured_in(unit:centimetre)].

Sarah
  isa person;
  height: 168 ~ [is_measured_in(unit:inch)].
```

**Fig. 2.** 2 topic maps to be merged

Both topic maps merged together would result in

```
Sarah
  isa person;
  height: 168 ~ [
    is_measured_in(unit:centimetre);
    is_measured_in(unit:inch)
  ].
```

**Fig. 3.** Accidental merge of 2 occurrences

and not in[2]

[2] In a hypothetical TMDM variant,“∼=” denotes a *defining annotation* (see below).

```
Sarah
  isa person;
  height: 168 ~= [is_measured_in(unit:centimetre)];
  height: 168 ~= [is_measured_in(unit:inch)].
```

**Fig. 4.** Hypothetical prevention of accidental merge of 2 occurrences

As it may intuitively be expected. This means that two separate statements (which may be complementary and not necessarily contradictory to each other) may be merged into one statement, where the annotation is contradictory or at least not intended to be merged.

This *accidental merge* effect, albeit a mostly theoretical problem, makes the annotation-style slightly unattractive. However, there may be other information regarding a quantity, such as the precision or uncertainty about that quantity, with needs to be represented somehow. It is hard to see that also precision or uncertainty information should be squeezed as another special theme into the scope of an occurrence[3].

The decision between the scope-style and the annotation-style is hard. Compactness and the absence of the accidental merge effect lean towards the scope-style, highest compatibility with implicit-unit (read: unitless) data sources and legacy topic maps as well as lower awkwardness lean towards the annotation-style. When also taking other information about a quantity into consideration, neither scope-style nor annotation-style are a perfect match. However, both styles are to be preferred over the other mentioned styles (Fig. 1, Fig. 4). The author leans slightly to the annotation-style, due to its non-intrusiveness.

### 2.4 Special TMDM support for units

In case the TMDM is to be changed any time far in the future, another option is to support references to unit topics by occurrences directly, similarly to the current support of references to type topics by occurrences (and names). In TMDM [ISO13250-2] language, occurrence items would have an additional `[unit]` property which is a topic item (or null). The topic referenced represents the unit of the quantity represented.

### 2.5 Special TMDM support for *defining annotations*

When considering both unit and precision information, it appears much more natural to attach this information to the occurrence, like information is attached to a topic as usual. Thus, reification of the occurrence could be employed. However, the usual reification carries the *accidental merge* problem, as explained above. To avoid this problem, it is imaginable to allow a second type of reifier for each statement, called *defining reifier*.

[3] Actually, scope is well-suited for representing uncertainty like “In 60 % of all cases, this exact quantitative statement is true.”. However, it is not that well-suited for representing precision, like “Actually, this value is not true, however it is 95 % probable that the true value is within $\pm 1$ % of the given value.”

A defining reifier is a topic which has certain statements just like other topics have. However, if two statements $S_0$, $S_1$ are to be considered equal by the standard TMDM equality rules, their defininig reifier needs to be equal, too. (Either both are null or both defining reifier topics exist and their sets of statements are equal[4].) This means if one statement of one defining reifier is slightly different than a corresponding statement of the other defining reifier (e.g. "`is_measured_in(unit:centimetre)`" vs. "`is_measured_in(unit:inch)`"), then $S_0$ and $S_1$ will still not be merged, because their defining reifiers define their identity. As their defining reifiers are not equal (in the sense of equal statement sets), $S_0$ and $S_1$ are not identical, and thus the *accidental merge* problem is prevented.

The beauty of this approach is its genericity. It is not limited to solve a particular use case (representing units of quantities in occurrences), but it allows the Topic Maps author to control in a much more fine grained way what constitutes the identity of a particular statement, and what not. However, this approach needs to be explored thoroughly, as side effects like merging of defining reifiers with other topics (and thus the defining reifier gaining possibly unwanted statements) need to be studied.

## 3 Representing units

How can units be represented as a topic map (such that other topic maps can reference these topics)? First, we observe that the *Système international d'unités* ("SI") [SI2006] is the currently predominant system of units of measurement. This makes it the best candidate to base a units ontology on[5]. (Non-SI-units can still be supported.) The SI defines *base units* (e.g. *metre*, *kilogram*, *second*, *ampere*, *kelvin*, *mole*, *candela*), *coherent derived units* ("products of powers of base units", e.g. $m^1 \cdot s^{-2}$), *coherent derived units with special names and symbols* (e.g. ohm ($\Omega$) for $m^2 \cdot kg \cdot s^{-3} \cdot A^{-2}$, tesla (T) for $kg \cdot s^{-2} \cdot A^{-1}$, newton (N) for $m \cdot kg \cdot s^{-2}$,...), *coherent derived units whose names and symbols include coherent derived units with special names and symbols* (e.g. coloumb per cubic metre ($C/m^3$) for $m^{-3} \cdot s \cdot A$,...), *SI prefixes* (e.g. kilo (k) for $10^3$, giga (G) for $10^9$, micro ($\mu$) for $10^{-6}$,...). The SI also accepts some non-SI-units (e.g. hour (h) for 3600·s, litre (L or l) for $10^{-3} \cdot m^3$,...). Finally, it defines *derived units* (e.g. milliwatt (mW) for $10^{-3} \cdot m^2 \cdot kg \cdot s^{-3}$, kilowatthour (kWh) (= 3 600 000 joule) for $3\,600\,000 \cdot m^2 \cdot kg \cdot s^{-3}$, ...). All these (different types of) units, prefixes and their interrelations as well as their names are to be represented in a topic map.

### 3.1 Scalers

One important use case for a units topic map is to convert a quantity in one unit (e.g. cubic metre per day) into a quantity in a different but compatible unit (e.g. litre per second).

[4] Equality of statement sets of two topics A, B shall be defined as: If both topics A, B were merged for some reason into a topic C, then the C would have the same number of statements as both A as well as B already have. Note that this definition is recursive, as the number of statements of C depends on merging rules, which in turn depends again on equality rules.

[5] However, see also the recently published, more comprehensive but also SI compatible standards set "International System of Quantities" [ISO80000].

As for each SI-compatible unit, we can find exactly one canonical coherent derived unit, it is sufficient to link each unit to its canonical coherent derived unit together with the appropriate conversion factor. This can be accomplished with a ternary association (see Fig. 5).

```
scaled_version(
  canonical_unit: cubic_metre_per_second,
  scaled_unit:    cubic_metre_per_day,
  scaler:         [
    scaler:1_per_86400;
    isa scaler:scaler;
    numerator:       1;
    denominator: 86400;
  ]
);
```

**Fig. 5.** A scaled-version association between a canonical coherent derived unit, a scaled unit and its conversion factor

Note that the conversion factor is in most cases a rational number. As rational numbers cannot be expressed in ternary associations directly, they are encapsulated in a topic of type *scaler*. Note that scalers are reusable (e.g. the conversion factor between $m^3/d$ and $m^3/s$ is the same as the conversion factor between kg/d and kg/s): When giving the scalers proper subject identifiers (e.g. scaler:1_per_86400), multiple instances of scalers will be merged together[6]. Note also that some scalers are actually SI prefixes and can thus get additional subject identifiers and additional names.

```
scaler:10000000
  isa scaler:scaler;
  numerator:   1000000;
  denominator: 1;
  scaler:mega;          # additional subject identifier
  - scaler:prefix_abbreviation: "M"; # SI prefix symbol
  - scaler:prefix_full: "mega";
  - scaler:prefix_full: "Mega"; @lang:de # German
  - scaler:prefix_full: "μεγα"; @lang:el # Greek
  - scaler:prefix_full: "מגה";  @lang:he # Hebrew
.
```

**Fig. 6.** A scaler which happens to be a SI prefix

[6] A side effect is that the scaler between inch and metre is not "`scaler:254_per_10000`" but "`scaler:127_per_5000`", because a canonical way of writing a fraction (e.g. its irreducible form) should be used, else the scaler merging effect is less likely to be exploited.

### 3.2 Autogenerating scaled units

A Ruby Domain Specific Language (DSL) and a corresponding DSL interpreter has been written to generate a topic map which represents units of measurement. The DSL instance starts with the definition of generic *scaler set*s (e.g. the set of all positive SI prefixes, the set of all SI prefixes, the set of all binary prefixes [IEC6007-2],...). Then, base units are declared (e.g. metre, second, ampere,...[7]). Each base-unit-declaration receives a generic scaler set and names of the base unit as parameter, as well as additional (non-SI-)units and their names if they are compatible.

```
base_unit si_prefixes, "s", :en_GB => "second",
                            :de    => "Sekunde" do
  other_scale ScalerSet.new {
    base
    multiply 60, "min", :en_GB => "minute",
                        :de    => "Minute"
    multiply 60,   "h", :en_GB => "hour",
                        :de    => "Stunde"
    multiply 24,   "d", :en_GB => "day",
                        :de    => "Tag"
  }
end
```

**Fig. 7.** Declaration of units for time. Units whose names are non-systematic (e.g. which cannot be composed by a prefix and a base unit name) are explicitly declared by unit-specific scaler sets

Once base units are declared, derived units can be declared, as simply as:

```
unit meter/second
unit meter/(second**2)
unit second**-1 do
  other_scale ScalerSet.new {
    base
    divide  1, "Hz", :en_GB => "Hertz",
                     :de    => "Hertz"
    divide 60, "RPM",:en_GB => "rounds per minute",
                     :de    => "Umdrehungen pro Minute"
  }
end
```

**Fig. 8.** Declaration of derived units m/s, m/s2, 1/s and all their scaled versions

[7] The particular implementation silently uses “gram” as internal base unit instead of “kilogram”.

Note that such a unit declaration not only creates a topic for the unit mentioned, but it also generates topics for all scaled version and the appropriate scaled_version associations. For example, the line "`unit meter/(second**2)`"[8] not only declares the topic unit:meter_per_square_second, but also the topics `unit:kilometer_per_square_day`, `unit:yard_per_square_hour`, and so on.

This autogeneration allows a topic map author to simply declare the "unit" prefix and then to refer to "`unit:how_it_would_be_spelled_out`" each time when actually referencing a unit of measurement.

The autogenerator also gives each generated unit appropriate names, such as:

```
unit:kilometer_per_square_day
- unit:symbol: "km/d²";
- "kilometer per square day" @lang:en-US;
- "kilometre per square day" @lang:en-GB;
- "Kilometer pro Quadrattag" @lang:de;
.
```

**Fig. 9.** Autogenerated names for units

The production of those names is dependent on the availability of language-dependent grammar modules. Such grammar modules are currently implemented for American English, British English, German[9], as well as unit symbols.

### 3.3 Autogenerating units with special names and symbols

Units with special names and symbols can be declared like base units:

```
special_name_unit si_prefixes, "W", :en_GB => "watt",
                                    :de    => "Watt"
```

**Fig. 10.** Declaration of a unit with a special name

Currently, however, the implementation of the unit topic map generator does not link the special unit to its coherent derived unit (which, in this particular case, would be `unit:square_meter_kilogram_per_cubic_second`), as this need did not arise so far.

[8] Note that American English spelling was used by the author. However, subject identifiers can be generated both for American English and British English spelling.

[9] The language topics employed here actually base on a currently unpublished language topic map which in turn is based on [BCP47] instead of directly ISO 639. The advantage of using BCP47 over other standards is that BCP47 language tags are precisely the language tags used by the HTTP protocol in its "Accept-Language" headers, facilitating simple processing in Topic Maps based web applications.

### 3.4 Non-SI base units

In some fields, other units are needed which are not expressable in SI units. For example, the unit "byte" ("B") as well as the unit "U.S. Dollar" ("USD")[10] are not expressable in SI units. Adding these units to our units ontology is not different to adding any other base unit. However, care must be taken, as there may be naming conflicts (such as the symbol "B" for "byte" as well as for "bel"[11]).

### 3.5 Scalability

The approach to autogenerate all scaled version of a coherent derived unit is efficient from the topic map author's point of view. Is a certain unit not represented, then only one line needs to be added in the DSL instance. However, there are scalability issues: if each base unit has 21 scaled versions[12] (e.g. yocto,..., nano,..., milli, centi, deci, (no prefix), deca, hecto, kilo,..., giga, yotta), then a coherent derived unit which builds on 2 base units has $21^2 = 441$ scaled versions (e.g. "kilometre per millisecond", "centimetre per hour",...). A coherent derived unit which builds on 5 base units has $21^5 = 4084101$ scaled versions (e.g. "square centimetre gram per square millisecond millikelvin kilomol"). Assume that data for each scaled version needs about 500 bytes when represented in a topic maps engine, then more than 2GB are needed to represent the scaled versions of just one coherent derived unit. Thus, a generic, downloadable "all units" topic map is too unwieldy to be widely usable. However, it is imaginable to create a dynamic web service which serves small pieces of information about a particular unit.

## 4 Future Work

While the autogenerated unit topic maps are usable and used in Topic Maps applications, there are still issues to be addressed.

First, a definite decision needs to be made which modeling style should be employed for quantities with units (e.g. scope-style or annotation-style, or maybe an entirely different style), perhaps by a standardization committee.

Second, the question when and how to link units with special names (e.g. "watt") to their derived unit (e.g. "square metre kilogram per cubic second") needs to be solved. Should both units merge, or not? The answer to this question has a deeper implication: As units are products of (integer) powers of base units, it is possible to represent this fact, too:

[10] It is questionable whether currency units are actually usable as measure (e.g. of monetary value) at all, as their values themselves are usually subject to change over time. Nevertheless, many users of currency units do not bother with this problem.

[11] A practical workaround employed by the author is to use "B" as a symbol for byte and to not use "B" as a symbol for "bel", but "B(A)" as a symbol for "bel (sound pressure level, A-weighted)" instead, as sound pressure measurements, where the unit "bel" is used often, are typically weighted differently for different frequencies (e.g. to match the weighting done by human hearing), and the most common weighting is the "A-weighting".

[12] Non-SI but compatible units like "hour" for "second" or "inch" for "metre" add on top.

```
product(power_2:       metre,
        power_1:       kilogram,
        power_minus_3: second) ~
  square_metre_kilogram_per_cubic_second;
```

**Fig. 11.** A unit as a reifier of a product of powers of base units

Representing units by reifiers of associations which represent the product of powers of base units is quite appealing. However, this concept as such cannot be extended to its fullest, as there are some side effects which may or may not be intended:

```
product(power_2:       metre,
        power_1:       kilogram,
        power_minus_3: second) ~
  watt;
```

**Fig. 12.** Another unit as a reifier of a product of powers of base units

As both associations (of Fig. 11, 12) are equal, they would merge. And due to this, the topics watt as well as `square_metre_kilogram_per_cubic_second` would merge. Thus, this product-reification-style implies an answer to the question whether both units are the same, or not. The answer is “yes”.

However, there may be some reason for the answer to be “no”, such as referencing the precise unit for display purposes. E.g., when the topic map author wants to refer to the unit “watt”, then the author may not want to refer to the unit “volt ampere” or the unit “joule per second” or to the unit “square metre kilogram per cubic second”. Thus, the product-reification-style prevents the topic map author from deciding this (but maybe this is good, because this decision should be up to the Topic Maps application instead).

Another problem is that this product-reification-style is limited to base units. It is not legal to state as a topic is allowed to reify at most one statement, but in this case, there are 2 statements to be reified simultaneously. Thus, expressing the relations between units in terms of non-base-units is not possible with this approach.

```
product(power_1:       joule,
        power_minus_1: second) ~ watt;
product(power_1:       volt,
        power_1:       ampere) ~ watt;
```

**Fig. 13.** The same unit as two different products of powers of other units

A third issue to be resolved is whether unit symbols should be part of another subject identifier of the unit they are a symbol for. In this case, it would be possible to refer to

a unit by "unit:W" or by "unit:m²kg/s³"[13], which is appealing. However, while the character '⁴' (U+2074) is part of the the characters which are allowed for a topic reference literal in the latest CTM draft [ISO13250-6-FDIS-2010-03], the characters '³' (U+00B3), '²' (U+00B2) as well as the character '/' (U+002F) are not. It may be trivial to include all superscript characters into the set of allowed characters, but the slash '/' may require careful attention, as it may have other meanings in present or future CTM. However, many URIs use slashes, and it may be imaginable that slashes should actually be part of the set of allowed characters for different reasons anyway. (Alternatively, as the intention of the slash '/' in this case is a fraction, the fraction slash '/' (U+2044), which is also not allowed presently, could be allowed instead of the slash '/', but the fraction slash '/' is quite awkward to type, which defeats the purpose of compactness.)[14]

A related question is about a canonical ordering of the powers of base units within a product of powers of base units. Is "unit:ampere_second" the same as "unit:second_ampere"? Should a derived unit have all possible names (and subject identifiers), or only one name (and subject identifier) according to a canonical ordering? The SI document [SI2006] itself follows a canonical ordering when referencing derived units[15], but using this ordering appears odd from time to time (e.g. "second ampere" is prefered by the SI over "ampere second"). As most units are composed of not more than 5 base units, there will be at most 5!=120 orderings per unit, so supporting all orderings is still bearable.

## 5 Conclusion

The support for units of measurement by the current Topic Maps standards is limited, and as it turns out, there is no definite and straightforward answer to how to uniformly represent units. This indicates a need for standardization, either by an official standard (be it a change of the TMDM or a standardized use of features of the current TMDM) or by a de-facto standard. This paper has outlined some issues which arise when attempting to standardize representation of units, both in case of a particular quantity with a particular unit and in case of a generic units ontology.

[13] or by "unit:m²kgs³"

[14] Another candidate for short names of units of measurement is [UCUM2010]. However, the UCUM requires even more characters than the suggestion by this paper (such as curly braces, round parentheses, square brackets), but it also supports a wider range of units and even so-called non-units. (However, some of these units , such as "milliequivalent per kilogram and 8-hour shift", are not "pure". Nevertheless, the decision of what "purity" is acceptable may be left to the user; a unit notation may remain agnostic about this decision.).

[15] The ordering is metre, kilogram, second, ampere, kelvin, mole, candela. However, if the unit is written as a fraction, then the units in the numerator are always written before the units in the denominator.

## References

MUO2009. Diego Berrueta, Luis Polo: "Measurement Units Ontology" (November 2008) http://idi.fundacionctic.org/muo/

UCUM2010. Gunther Schadow, Clement J. McDonald: "Unified Code for Units of Measure" (April 2010) http://aurora.regenstrief.org/~ucum/ucum.html http://aurora.regenstrief.org/∼ucum/ucum.html

Leal2002. David Leal, Andrea Schröder: "RDF vocabulary for physical properties, quantities and units" (August 2002) http://www.s-ten.eu/scadaonweb/NOTE-units/2002-08-05/NOTE-units.html

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/

XMLSchemaPart2. World Wide Web Consortium: "XML Schema Part 2: Datatypes Second Edition" (October 2004) http://www.w3.org/TR/xmlschema-2/

Garshol2008Scope. Lars Marius Garshol: "A Theory of Scope" in *Scaling Topic Maps, Third International Conference on Topic Maps Research and Applications, TMRA 2007, Leipzig, Germany, October 11-12, 2007, Revised Selected Papers.* (2008) (ISBN 978-3-540-70873-5), pages 74-85 http://www.garshol.priv.no/download/text/a-theory-of-scope.pdf

ISO80000. International Organization for Standardization – Joint Technical Committee 12/International Electrotechnical Commission — Technical Committee 25: "ISO/IEC IS 80000 Quantities and units" International Organization for Standardization, Geneva, Switzerland (November 2009)

SI2006. Bureau International des Poids et Mesures: "Le Système international d'unités" (April 2006) http://www.bipm.org/utils/common/pdf/si_brochure_8.pdf

IEC6007-2. International Electrotechnical Commission: "IEC IS 60027-2:2005: Letter symbols to be used in electrical technology — Part 2: Telecommunications and electronics" International Organization for Standardization, Geneva, Switzerland (August 2005)

ISO13250-6-FDIS-2010-03. International Organization for Standardization/International Electrotechnical Commission – Joint Technical Committee 1 – Subcommittee 34 – Working Group 3: "ISO/IEC FDIS 13250-6: Information Technology – Document Description and Processing Languages – Topic Maps – Compact Syntax" International Organization for Standardization, Geneva, Switzerland (March 2010) http://www.isotopicmaps.org/ctm/ctm.html

# Subject Headings make information to be topic maps

Motomu Naito

Knowledge Synergy Inc., Takahama, Aichi, Japan,
motom@green.ocn.ne.jp,
http://www.knowledge-synergy.com/index-en.html

**Abstract.** This paper reports the efforts to make topic maps from Subject Headings (SHs) and discuss practical use of them for organizing information and knowledge. SHs are usually maintained by libraries and used in bibliographic records. SHs are thesauri and they are well organized. Fortunately some SHs are published on the Web. We transformed them to topic maps. Usually each subject in SHs has own ID. It can play PSI role. By keeping the relationships included in SHs such as Broader–Narrower, Related, Use–Use for etc in topic maps, information or knowledge can be linked together and organized according to the structure of SHs. In other words, by using SHs information and knowledge can be topic maps easily.

## 1 Introduction

Wikipedia redirects "Subject heading" to "Index term" and define the term as "An index term, subject term, subject heading, or descriptor, in information retrieval, is a term that captures the essence of the topic of a document. Index terms make up a controlled vocabulary for use in bibliographic records." [6] As well known Subject Headings, there are Library of Congress Subject Headings (LCSH) [8], Canadian Subject Headings (CSH), Medical Subject Headings (MeSH), etc. There are also National Diet Library Subject Headings (NDLSH) [7] and Basic Subject Headings (BSH) in Japan. They are being managed and maintained by time-consuming works of many people. SHs are thesauri [4]. They consist of subject headings, relationships among subject headings, scope note, etc. As relationships there are Broader–Narrower (BT–NT), Related (RT), Use–Use for (USE–UF) etc. Usually each subject heading has own ID that can be used as PSI (Published Subject Identifier).

As a part of the activities of Center for Integrated Area Study (CIAS) [9] in Kyoto University, we converted NDLSH, BSH and LCSH to topic maps and developed their web applications. SHs have fairly simple structure and are able to be converted to topic maps easily. They have a high affinity for each other and the topic maps can also inherit the relationships among subject headings and their IDs. We are trying to use those topic maps for linking and organizing information and knowledge.

In this paper we report our effort to make topic maps from Subject Headings. We also discuss how information and knowledge to be topic maps using them. In section 2, available Subject Headings are given. In section 3, practical uses of Subject Headings are discussed. In section 4, challenges we faced described. Finally conclusion and future work are shown in section 5.

## 2 Subject Headings and Topic Maps

In this section, we introduce available SHs on the web. We converted those SHs to topic maps. We explain the detail of those SHs and topic maps.

### 2.1 NDLSH

NDLSH (National Diet Library Subject Headings) has been maintained by National Diet Library (NDL) in Japan. We were provided NDLSH 2008 version data that were Tab Separated Value (TSV) format and we converted it to topic map. NDL opened web site for NDLSH in July 2010. We will use the latest data that can be downloaded from the site next time. Figure 1 shows ontology of the topic map. Topic types are represented by squares and association types are represented by lines. There are two topic types namely Subject Heading and Reference and three association types namely BT–NT, RT and Use–Use for (USE–UF). Interesting thing in the NDLSH is USE–UF relation between NDLSH and LCSH.

Table 1 shows topic types, association types and numbers of their instances.

We developed a web application based on the topic map using Ontopia [3]. We can navigate NDLSH topic map according to subject headings and references and relationships among them. The application has the following feature:

- instance topic list display
- topic detail display
- full text search
- tolog query

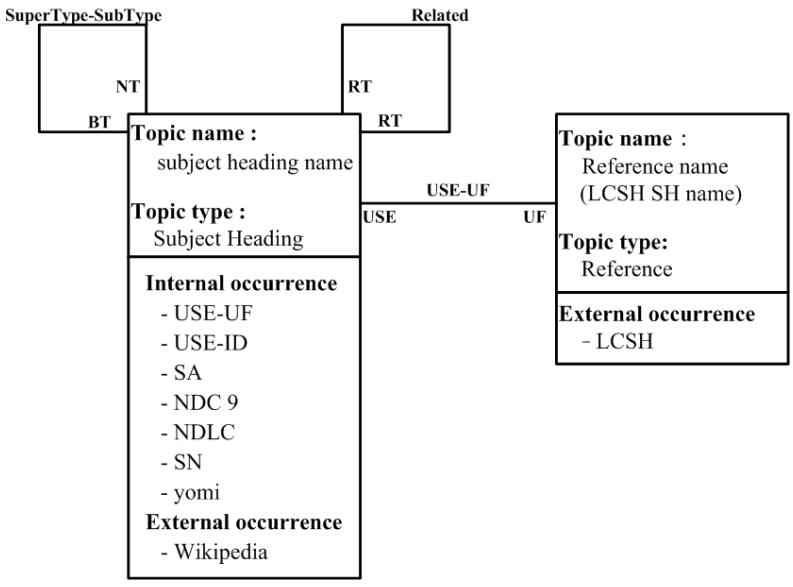



**Fig. 1.** Ontology of NDLSH topic map

**Table 1.** The number of types and instances

| Type | Type name | The number of instances |
|---|---|---|
| Topic type | Subject Heading | 17,953 |
| | Reference | 47,816 |
| Association type | Broader–Narrower | 13,220 |
| | Related | 9,738 |
| | Use–Use for with LCSH | 11,663 |

– graphic display
– link to LCSH
– link to Wikipedia

### 2.2 LCSH

LCSH has been maintained by Library of Congress in the US. We downloaded RDF/XML format of LCSH and converted it to a topic map using Omnigator included in Ontopia. Data size of LCSH RDF is huge. It is more than 400MByte. LCSH includes many elements. But because of the limitation of machine resource we extracted limited elements. We had to omit some elements such as "skos:altLabel", "owl:sameAs", "skos:closeMatch", etc. "skos:altLabel" could be converted to variant. "owl:sameAs" and "skos:closeMatch" could be linkage to another vocabulary system. Table 2 shows the extracted elements and their number.

Figure 2 shows ontology of the topic map. Topic types are represented by squares and association types are represented by lines. In the topic map there is only one topic types namely "core:Concept" and two association types namely "BT–NT" and "RT".

Table 3 shows topic type, association types and numbers of their instances.

We also developed a web application based on the topic map using Ontopia. We can navigate LCSH topic map according to subject headings (core:Concept) and relationships among them. The application has the following feature:

– instance topic list display
– topic detail display

**Table 2.** Extracted elements and the number of them

| Element | The number of elements |
|---|---|
| rdf:Description | 380,123 |
| rdf:type | 380,123 |
| skos:broader | 254,651 |
| skos:prefLabel | 380,110 |
| skos:related | 11,137 |
| skos:scopeNote | 11,482 |

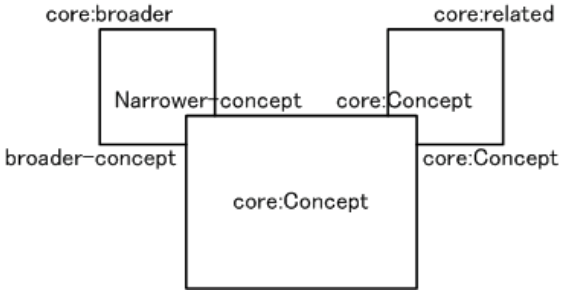



**Fig. 2.** Ontology of LCSH topic map

**Table 3.** The number of types and instances of LCSH

| Type | Type name | The number of instances |
|---|---|---|
| Topic type | core:Concept | 380,123 |
| Association type | Broader–Narrower | 254,651 |
| | Related | 11,137 |

- full text search
- tolog query
- graphic display

## 3 Practical use of Subject Headings

In this section we discuss practical uses of SHs. We tried some of them experimentally and we will try others in the future.

### 3.1 Organizing information according to SHs

By linking subjects in information to subjects in SHs, information can be organized according to SHs. Experimentally we organized Wikipedia's articles according to NDLSH. We can easily make the address of each Wikipedia's article. We can make it as follows:

```
"http://ja.wikipedia.org/wiki/" + "Beer"
```

In the above, "http://ja.wikipedia.org/wiki/" is constant and "Beer" is subject heading. We tried this all subject headings in NDLSH (the number is 17,953) and found 12,051 corresponding articles in Wikipedia. We used the articles as occurrences of corresponding subject headings. Figure 3 shows the organization of Wikipedia's articles.

Each article in Wikipedia is made from the bottom up and relationships among articles are not organized enough. But by using SHs, we could organize and navigate the Wikipedia's articles according to SHs.

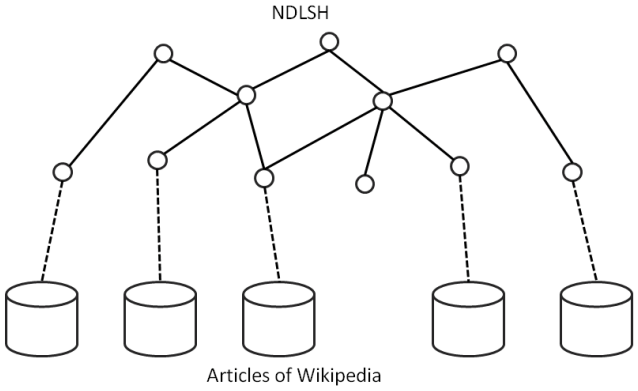



**Fig. 3.** Organizing Wikipedia

### 3.2 Multilanguage Mapping Using LCSH as a Core System

Many subject headings in NDLSH refer to subject headings in LCSH as UF reference. Keeping the reference as association in topic map, cross-reference between NDLSH and LCSH, in other words cross-reference between Japanese and English is made possible. First we converted NDLSH to a topic map that inherited the UF reference to LCSH. At

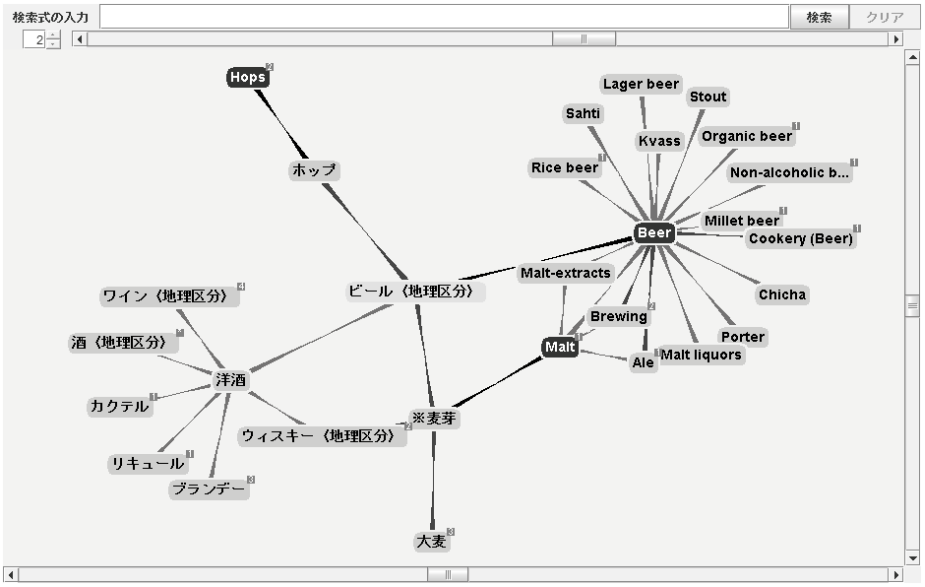



**Fig. 4.** NDLSH-LCSH mapping

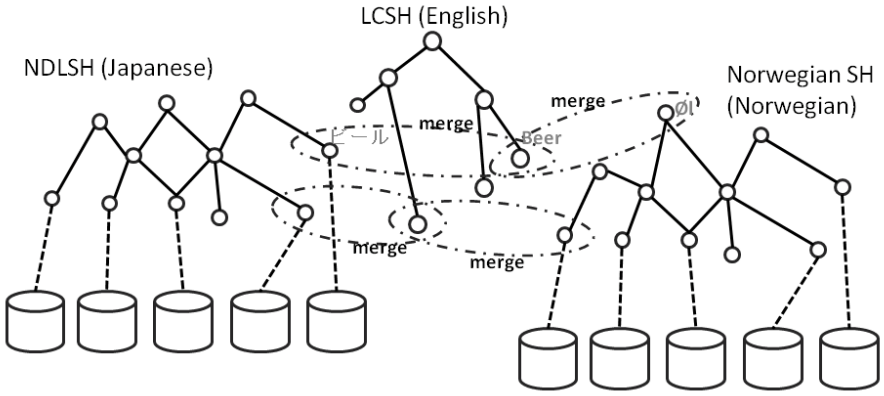



**Fig. 5.** Japanese Norwegian mapping

that stage we could refer from NDLSH's subject headings to corresponding LCSH's subject headings. Next we converted LCSH to a topic map and merge NDLSH topic map and LCSH topic map. UF reference in NDLSH combined with corresponding LCSH subject headings. In result we could map subject headings between NDLSHand LCSH and navigate from NDLSH to LCSH and vice versa. Figure 4 shows the merge result of NDLSH and LCSH. In the Fig. 4, left side nodes come from NDLSH and right side nodes come from LCSH. "ビール" and "Beer", "麦芽" and "Malt" and "ホップ" and "Hops" are being mapped each. In the same way, if cross-references are made between English and other languages, it is possible to make cross-references of subject heading during multi-languages. Figure 5 shows Japanese Norwegian mapping via LCSH.

### 3.3 Mutual Complementing of Our Concept Classification and SHs

The purpose of SHs is mainly to manage publication in Library. Usually they have rich subjects for general things. But they don't have enough subjects for specialized field if there are few publications about them. Meanwhile it is relatively easy for us to classify concepts our own specialized field. But classifying general things is very time-consuming work and it is almost impossible for us to do it with limited man power. We think mutual complementing of concept classification of specialized field and general field is very effective and reasonable. By making concept classification of our specialized field and merging them with SHs, we can get rich and comprehensive subject system. Fig. 6 shows mutual complementing our concept classification and NDLSH.

### 3.4 Other Uses

Moreover many other uses are considered. Several examples are given.

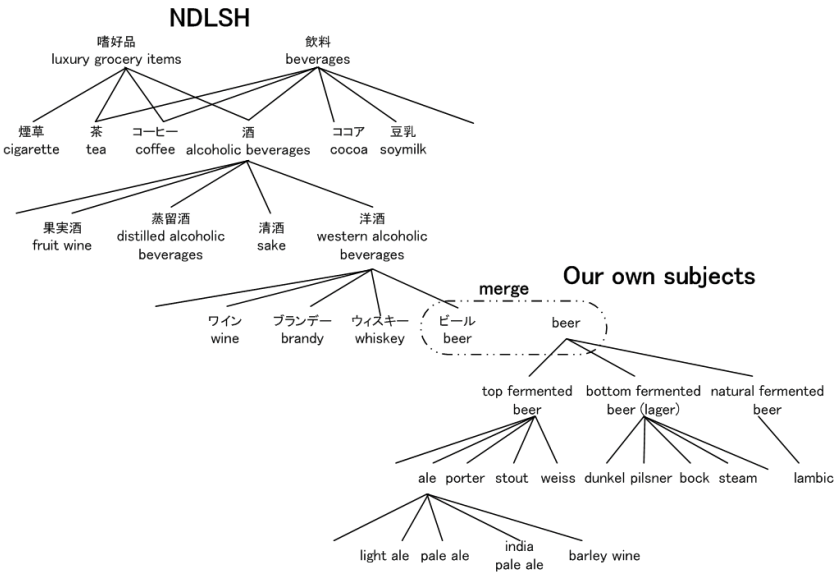



**Fig. 6.** Mutual complementing of our concept classification and SHs

**Using Subject Headings as PSI** Usually each subject heading has its own ID. We can make up a URI based on the ID and use it as PSI. Subject headings have many relationships among them, some relationships construct tree structure and other relationships construct network of subjects. We can keep those relationships in PSIs and exploit them. Actually, LCSH has the following form of URI for each subject (concept):

```
http://id.loc.gov/authorities/sh85012832#concept
```

Web version of NDLSH has also has the following form of URI for each subject.

```
http://id.ndl.go.jp/auth/ndlsh/00560674
```

**Subject Heading Providing Web Service** Each subject heading in Topic Maps has some characteristics i.e. names, associations and occurrences. If each subject heading has PSI, it becomes possible to offer web service that provides those characteristics [10,12,13]. Users can ask the service about given subject with PSI and the service returns the characteristics about the subject using protocols such as TMRAP [11,14,15], SDShare, etc.

**Using SHs as Test Data** Anybody can download LCSH and NDLSH from the Web. Their sizes are fairly big. For example, there are more than 380,000 subjects and 254,000 BT–NT relations in LCSH. The structures of them are simple and easy to understand. We can convert them to topic maps easily. We can use them as test data for Topic Maps engine, Query engine, etc.

# 4 Challenges

## 4.1 Attach or Extract Subjects from Information

We have already a huge amount of information that is wanted to be topic maps. Some are made by us and some are external information. By linking subjects i.e. topics in SHs topic maps to the subjects in information, we can organize those information according to the SHs. In the case of subjects are attached to information, we can use them. But in the case of no subjects are attached to information, a big challenge is how to attach subjects to information or extract subjects from information. We organized Wikipedeia experimentally. But the case is not subject base matching but literal base matching. In order to deal with this challenge two ways are considered. The first one is to attach subjects to information by human. To do so some tagging systems are required. The second one is to extract subjects from information automatically. In that case subject extraction tools are required.

## 4.2 Large Data

We converted LCSH to topic maps. Original format of LCSH is RDF/XML. The size is very large, more than 400MByte. We used the machine that had 4GByte memory. When we tried to convert it to topic map, "Out of memory" error occurred frequently. At the moment we could convert only limited elements. We had to select some element and we had to neglect other elements. We also divided the file into several files and merge them after conversion. If we want to merge other topic maps, the situations become more severe. We need scalable and stable environment to handle big files.

# 5 Conclusion and Future Work

We have already stored huge amount of information that want to be topic maps. Many well organized knowledge has already existed. SHs are one of them. We converted LCSH and NDLSH to topic maps. Experimentally we tried to organize Wikipediea articles using NDLSH topic map. We also tried to realize Japanese–English mapping by merging NDLSH with LCSH. We found SHs very useful and effective to organize information. There are many practical way to use them as described in section 3.

We will continue to work with SHs in the future:

- To try our information to be topic maps according to SHs.
- To try to achieve multi-language mapping based on SHs.
- To try to find out the good ways to link subjects in information to their corresponding subjects in SHs.
- To try to realize the web service that provides subject headings and their characteristics.

## References


1. Motomu Naito: Topic map for Topic Maps case examples, Fourth International Conference on Topic Maps Research and Applications, TMRA 2008 Leipzig, Germany, October 16–17, 2008. pp 261–272
2. 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
3. Steve Pepper: As We REALLY May Think : Memex, Topic Maps, and subject-centric computing, http://www.ontopedia.net/pepper/slides/AToMS2007.ppt
4. Lars Marius Garshol: Metadata? Thesauri? Taxonomies? Topic Maps: Making sense of it all!, http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html
5. Ontopia: Tools for building, maintaining, and deploying Topic Maps-based applications, Open Source, http://code.google.com/ontopia/
6. Wikipedia: Index term, http://en.wikipedia.org/wiki/Subject_heading
7. National Diet Library: National Diet Library Subject Headings (NDLSH), http://id.ndl.go.jp/auth/ndlsh
8. Library of Congress: Library of Congress Subject Headings (LCSH), http://id.loc.gov/authorities http://id.loc.gov/authorities
9. CIAS: Center for Integrated Area Studies, Kyoto University, http://www.cias.kyoto-u.ac.jp/english/CIAS/index.html
10. Motomu Naito: Topic Maps Web Service: Case Examples and General Structure, Fifth International Conference on Topic Maps Research and Applications, TMRA 2009 Leipzig, Germany, November 12–13, 2009. pp 179–184
11. Ontopia: TMRAP : Topic Maps Remote Access Protocol, http://www.ontopia.net/topicmaps/tmrap.html/
12. Lars Marius Garshol: tmphoto : Lars Marius's photos, http://www.garshol.priv.no/tmphoto/
13. Motomu Naito: tmcase1 : Topic Maps case examples, http://www.garshol.priv.no/tmcase1/
14. Lars Marius Garshol: TMRAP support in the blog, http://www.garshol.priv.no/blog/145.html
15. Lars Marius Garshol: The get-illustration web service, http://www.garshol.priv.no/blog/183.html

# Inquiry Optimization Techniquefor a Topic Map Database

Yuki Kuribara[1] and Masaomi Kimura[2]

[1] Shibaura Institute of Technology,
Department of Electrical Engineering and Computer Science
3-7-5 Toyosu, Koto-ku, Tokyo 135-8548, Japan
[2] Shibaura Institute of Technology,
Department of Information Science and Engineering
3-7-5 Toyosu, Koto-ku, Tokyo 135-8548, Japan

**Abstract.** In this paper, we introduce the inquiry optimization technique for a topic map database. We assumed a tolog-type query language and the topic map data structure defined by TMDM (ISO/IEC 13250 part-2). We show a strategy for making a retrival plan based on the typical size of topic map objects and the number of objects to be loaded in memory. As for implementation, we focused on one of the queries that has multiple plans, the type of query that retrieves topics related to a certain topic by some relationship both of which are specified in the query. We proposed an esimation formula for the retrival cost of each route and evaluated our technique by applying it to the topic map database called TOME, which is a prototype developed by the authors.

## 1 Introduction

There are some topic map database systems that support queries to topic maps. However, when we retrieve information on a topic map, optimization of inquiry is important for the database, since we need a long retrieval time if the target topic map is large. Despite its importance, existing topic map databases seem not to take this into consideration. As far as the authors have investigated previous studies, ways to optimize inquiry have not been studied.

In this paper, we introduce the execution plan optimization technique for queries submitted to a topic map database.

We assume a tolog-type[1] query language and the topic map data structure defined by the topic maps data model (ISO/IEC 13250 part-2) [2]. We also assume an object-oriented database (OODB) as a container for topic map objects such as *topic maps*, *topics*, *associations*, and so on. Our container is not a relational database. Since it defines relationships among objects only in an implicit way, not explicitly as done by OODB, taking advantage of the structures in a topic map is unsuitable. The authors have developed a prototype topic map database TOME[3], which utilizes the open-source OODB, *db4o*[4].

We first show a strategy for making a retrival plan based on the typical size of topic map objects and the number of objects to be loaded in memory. As for implementation, we focus on one of the queries that has multiple plans, the type of query that retrieves

topics related to a certain topic by some relationship both of which are specified in the query. Next, we propose an esimation formula for the retrival cost of each route. Last, we evaluate our technique by applying it to the topic map database TOME.

## 2 Optimization Technique Based on Object Reference Relationships Defined by the Topic Maps Data Model

### 2.1 Prerequisite

We assume that the data structure of topic maps complies with the topic maps data model (TMDM) defined in ISO/IEC 13250 part-2. The data model consists of seven types of *information items* and 19 types of *named properties* in topic maps. We utilize the relationships between topic map objects, namely, the reference relationships between objects such as topics, association roles, and associations.

In *TOME, the relationships among the classes in OODB, denoting topics*, *associations*, and so on, are designed to conform to the data model defined by TMDM. The objects in TOME are instances of information items and named properties.

As for the query language, we assume that it contains similar kinds of queries defined by *tolog, which is the query language originally proposed by Ontopia and supported by OKS and* TM4J. In a later section, we show the cost estimation formula for the query that is equivalent to that defined in tolog.

### 2.2 Retrieval Routes Induced by TMDM

Figure 1 shows the main topic map objects defined as information items and their reference relationships.

As we can easily see in the figure, we can retrieve objects that have a connection to some specified object by means of the reference relationships, e.g., we can retrieve some association referred by a topic map, and can also retrieve topics referred by the association roles that are related to the association. It is also easy to see that we can retrieve topics directly from the topic map. This suggests that there may be a number of ways to retrieve the same topic map objects.

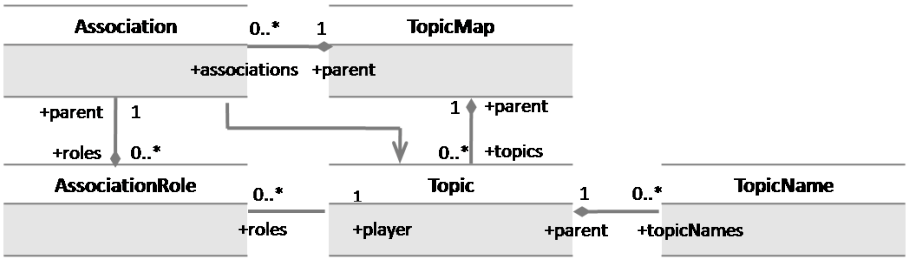



**Fig. 1.** Main topic map objects and their reference relationships defined by TMDM

As relational databases are designed, topic map databases should have the responsibility of identifying an ideal query execution plan. This has to be emphasized if its query language is a type of tolog, since it does not specify the query execution plan, as it does in the case of SQL for relational databases.

### 2.3 The Multiple Plans of the Query that Retrieves Topics

Let us illustrate the multiple plans for the query that is based on that defined in tolog. In this section, we introduce the scenario of retrieving topics that are related to some topic by the given associations. We focus on the scenario, since it is the most important to retrieve the topics that are related to the known topic.

Figure 2 shows the retrieval plan for topic objects, by which we search association objects specified in a query first and then topic objects that are connected to them (the association route). In fact, there is another plan to retrieve topic objects that are connected to association objects, which is shown in Fig. 3 (the topic route). It shows the plan that retrieves the topic objects first and then finds the intended topic objects by choosing those that are connected to the association objects specified in a query. In fact, tolog supports this kind of query, and we consider that it is mainly used since we usually search topic objects based on relationships (associations) with other topic objects.

There is a difference among the costs of these plans, namely, the number of topic objects and association objects to be loaded from a storage device into memory. Obviously, this indicates a difference in performance. In order to have the database management system process queries effectively, we must design an estimation method for the cost of retrieving the objects. We propose an estimation formula for retrieval of objects corresponding to each of these figures.

As for other query syntaxes, we also look into the retrieval plans for 20 query syntaxes defined in tolog. The syntaxes include the retrieval syntaxes of topics from a topic map, the retrieval syntax of associations of type and instance, and so on. As a result, we found

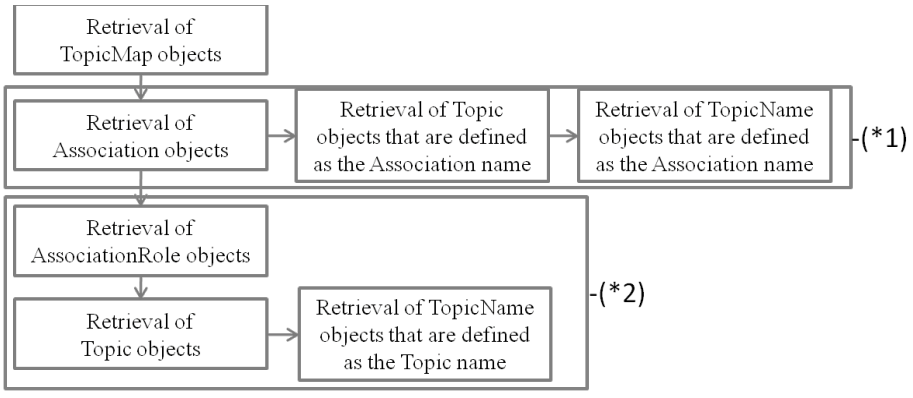



**Fig. 2.** Retrieval plan for topics (the association route): retreiving associations first and topics next

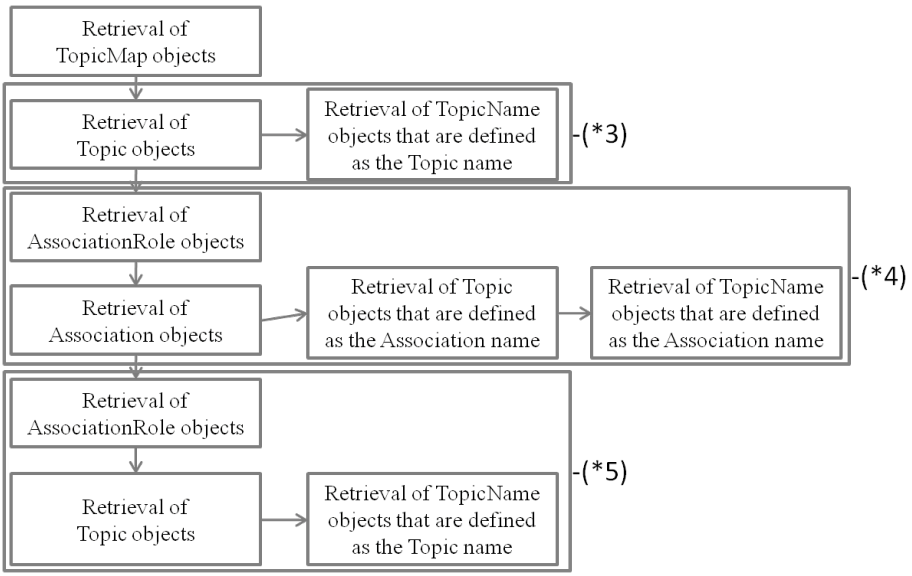



**Fig. 3.** Retrieval plan for topics (the topic route): retrieving topics first and then restricting them by matching names of associations

that the system can deal with information of a topic map more efficient by choosing a suitable plan for the query syntaxes, ‘association-role’, ‘role-player’, ‘item-identifier’, ‘instance-of’, ‘type’ and ‘direct-instance-of’.

## 3 Cost Estimation of Retrieval

In order to choose a suitable plan for processing queries, we introduce an estimation formula for the retrieval cost of each plan. Although there are more than 20 query syntaxes in tolog, in this section, we focus on the syntax that supports the query introduced in the last section. Our target query is to retrieve topic objects that are referred by a specific association with a particular topic specified in the query.

Our estimation strategy is similar to that for cost estimation for processing SQL queries submitted to relational databases. The different points come from the structures of the target data. The target data in relational databases are tabulated, and each record has the same number of data corresponding to each other. However, our target data have a more complex structure; the number of correspondences in data, such as the correspondence between topics and associations, and the size of the object can vary depending on the object. This therefore makes the cost estimation formula more complex than that for relational databases.

### 3.1 Retrieval Cost Estimation of Objects Compared to the Whole Cost

Before we define the cost estimation formulae, we measured the total execution time and the retrieval time of each object, and estimated how the retrieval of objects dominates the whole cost to process a query.

**Table 1.** Comparison of object retrieval time with the whole execution time

| | Execution Time (A) (nano sec) | Retrieval time of objects (B) (nano sec) | Other instruction time (A-B) (nano sec) | The ratio of object retrieval time (B/A) (%) |
|---|---|---|---|---|
| Association Route | $6.025 \times 10^8$ | $5.991 \times 10^8$ | $3.40 \times 10^6$ | 99.44 |
| Topic Route | $1.035 \times 10^8$ | $1.033 \times 10^8$ | $2.0 \times 10^5$ | 99.81 |

Table 1 shows the whole execution time (A), retrieval time of objects (B), the time obtained by A-B, which shows the execution time other than object retrievals, and the ratio of object retrieval time B/A.

This result shows the object retrieval time (B) dominates the processing time (A) more than 99% for both routes. This indicates that it is enough to measure the time to retrieve objects to evaluate the cost of query processing. Based on this finding, we define the cost estimation formulae that estimate the retrieval time of objects.

### 3.2 Assumption about the Retrieval Cost of Objects

We identify the retrieval cost of objects according to their size. We assume each object size can be represented by a mean value in each class. Of course, the size of each object can differ. However, the unit of data is a serialized object, which includes not only field data but also methods. The reason for our assumption is that the size of methods regarded as data exceeds that of the field data. The size of methods is unique to the class to which the object belongs. Moreover, typical data in the fields are the names of objects such as topics, associations, and so on. The size difference in names will be much smaller than the size of the object itself and may be negligible.

It is also reasonable to assume that the size of objects differs if they belong to different classes, since each class has different methods.

In the following, the number of association objects in the topic map is denoted as N, the number of topic objects, as M, and the number of the unique association names, as Q. The retrieval cost of association objects, topic objects, topic name objects, and association role objects is denoted as $C_a$, $C_t$, $C_{tn}$ and $C_{ar}$, respectively.

### 3.3 Cost Estimation Formula for the Association Route

Let us define the formula for estimating the cost of the association route. The route can be decomposed into two parts denoted by (*1) and (*2) in Fig. 2. In part (*1), we retrieve association objects, topic objects that express their names, and their topic name objects. In part (*2), we retrieve the association role objects referred by the associations, the topic objects corresponding to the association roles, and their topic name objects.

For part (*1), we must estimate the number of association objects, the topic objects, and the topic name objects that should be loaded from a storage device into memory.

Let us discuss the number of association objects to be retrieved. Remember that TMDM permits the redundant existence of multiple associations that have the same name in a topic map. In order to find all associations to be retrieved, we need to retrieve all association objects in the topic map.

The cost of topic objects and topic name objects is given by multiplying the cost of a single object of each class by the number of retrieved association objects. Here, note the effect of buffers. If the same objects have been loaded into memory, they are buffered in memory. Since this shortens the retrieval time, the cost can be reduced by multiplying the coefficient:

$$\alpha = \frac{Q}{N} + r\left(1 - \frac{Q}{N}\right) . \tag{1}$$

Here, $r$ is the effective retrieval ratio of cost loading from a storage device to the cost reading of the buffer.

For (*2), the expected number of association role objects is given by $N/Q$, if we assume that the association roles are uniformly assigned to associations. We retrieve topic objects that are related to the association role objects and their topic name objects as many times as the number of association roles. Again, buffering should be taken into consideration for topic objects and topic name objects, whose coefficient is given as:

$$\beta = \frac{M}{2N} + r\left(1 - \frac{M}{2N}\right) . \tag{2}$$

Taking these aspects into consideration, we define the cost estimation formula for the association route as follows:

$$C_1 = \big[C_a + \alpha(C_t + C_{tn})\big]N + \big[C_{ar} + \beta(C_t + C_{tn})\big]\frac{N}{Q} . \tag{3}$$

### 3.4 Cost Estimation for the Topic Route

As we did in Section 3.2, we decompose the route shown in Fig. 3 into three parts. In the part denoted as (*3), we retrieve the topic object that has the topic name given. Remember that TMDM permits the existence of the only one topic that has the same name. The result is that the expected number of steps for searching topics is given by half the number of topic objects.

In part (*4), we retrieve the association role objects and the association objects that are referred by the topic objects obtained in part (*3). In order to refer the name of the association, we also have to load the topic objects referred by the associations and their topic name objects. As we did in the last section, taking buffering into consideration, we multiply coefficient $\alpha$ by the cost of each of them. The number of association role objects is estimated to be twice the number of associations per topic. (Regarding the topic map as a graph in graph theory, this is equal to the average degree.)

In part (*5), we retrieve the association role objects referred by the associations obtained in part (*4). The expected number of steps for searching the association roles is given by $2N/MQ$, which denotes the average number of associations that have the

name specified by the query. For each of the association objects obtained in part (*4), the cost of retrieving the association role objects, the topic objects, and the topic name objects is given as it was in the last section.

The resultant cost estimation formula for the topic route is given as follows.

$$C_2 = (C_t + C_{tn})\frac{M}{2} + \left[C_{ar} + C_a + \alpha(C_t + C_{tn})\right]\frac{2N}{M} + \left[C_{ar} + \beta(C_t + C_{tn})\right]\frac{2N}{MQ}. \quad (4)$$

Comparing cost $C_1$ and $C_2$, we can choose the better plan for retrieving the target topic objects.

## 4 Experiment

In order to demonstrate our method of optimizing a query execution plan, we applied our technique to topic maps in our database prototype system, TOME. As targets, we selected two topic maps that have different sizes. The target topic maps were the Rampo Edogawa topic map and the Pokemon topic map [5]. The Rampo Edogawa topic map includes topics about Rampo Edogawa, who is a famous mystery story writer in Japan, his works, and his hometown. The Pokemon topic map includes Pokemon names and their attributes as topics, and evolutional and attribute relationships as associations. The Rampo Edogawa topic map has 22 topics, 56 topic names, 15 associations, and 30 association roles, and the Pokemon topic map has 174 topics, 174 topic names, 432 associations, and 864 roles. Figure 4 shows the structure of the Pokemon topic map.

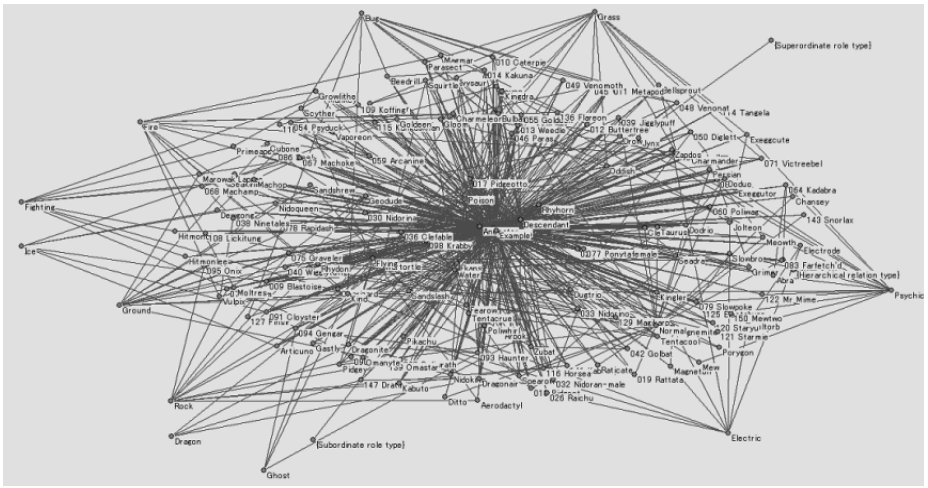



**Fig. 4.** Structure of the Pokemon topic map

**Table 2.** Cost estimation of an object of each class

| Topic Maps | The object name | The retrieval time (nano sec) | The normalized value by setting the retrieval time to be 1 | The object Size (byte) | The normalized value by setting the object size to be 1 |
|---|---|---|---|---|---|
| Rampo Edogawa Topic Map | The retrieval time of topic | 969200 | 3.343 | 608 | 4.75 |
| | The retrieval time of topicname | 496700 | 1.713 | 376 | 2.938 |
| | The retrieval time of associationrole | 289900 | 1 | 128 | 1 |
| | The retrieval time of association | 562600 | 1.940 | 376 | 2.938 |
| Pokemon Topic Map | The retrieval time of topic | 1053000 | 5.501 | 608 | 4.75 |
| | The retrieval time of topicname | 501600 | 2.621 | 376 | 2.938 |
| | The retrieval time of associationrole | 191400 | 1 | 128 | 1 |
| | The retrieval time of association | 577700 | 3.019 | 376 | 2.938 |

### 4.1 Estimation of Cost of Retrieving a Single Object of Each Class

We start by estimating the retrieval cost of association object $C_a$, topic object $C_t$, topic name object $C_{tn}$, and association role object $C_{ar}$. Table 2 shows the retrieval time and the size of a single object for each class. We measured the mean execution time of the queries for the topic retrieval repeated 174 times, the topic name retrieval repeated 174 times, the association retrieval repeated 432 times, and the role retrieval repeated 864 times. Since the ratio of these costs is essential in our study, we proportionally normalize the values of cost by setting the retrieval time and the object size as 1. Regardless of the difference in topic maps, we can see a similar tendency between the retrieval time and the object size. We therefore adopt the ratios of object sizes as the values of cost. Namely, $C_a$ and $C_{tn}$ are 2.94, $C_t$ is 4.75, and $C_{ar}$ is 1.00.

### 4.2 Estimation of Cost for each Query Execution Plan

In order to evaluate our cost estimation formula, we measured the execution time of three queries for each topic map and compared the tendency of the value of cost. In TOME, we implemented programs to measure the time elapsed from query start to end for the two query execution plans shown in Sects. 3.2 and 3.3.

Table 3 shows the resultant data of the average query execution time that we measured ten times and the evaluated cost for each query execution plan and the ratio (the association route/the topic route) of time and cost. From this figure, both for the execution time and for the estimated cost, we can see the tendency that it is effective to execute the association route for the Rampo Edogawa topic map and the topic route for the Pokemon topic map, as far as the order of magnitude of the ratio is concerned.

The reason for the errors may originate from the simplification of our assumption. Designing the estimation formula, we assumed a similar condition to estimate the cardinality of tables in relational databases, namely the condition in which the distribution

**Table 3.** Query execution time and estimated cost

| Topic Maps | Specified name (Topic/ Association) | The query execution time (nano sec) | | | The evalueated cost for each query execution plan | | |
|---|---|---|---|---|---|---|---|
| | | The association route | The Topic route | The ratio of time (Association/ Topic) | The association route | The topic route | The ratio of cost (Association/ Topic) |
| Rampo Edogawa Topic Map The topics : 39 The associations : 15 | Rampo/ Born-in | 31 | 157 | 0.1975 | 133.2 | 164.0 | 0.8125 |
| | Rampo/ Write | 47 | 187 | 0.2513 | 133.2 | 164.0 | 0.8125 |
| | Kogoro/ appear | 78 | 203 | 0.3842 | 133.2 | 164.0 | 0.8125 |
| Pokemon Topic Map The topics : 174 The associations : 432 | Picachu/ evolve | 297 | 31 | 9.581 | 2533 | 697.7 | 3.630 |
| | Normal/ reside | 297 | 156 | 1.904 | 2533 | 697.7 | 3.630 |
| | Dewgong/ evolve | 249 | 125 | 1.992 | 2533 | 697.7 | 3.630 |

of the number of data that have the same value is uniform. We assumed this condition because of the performance of computation. If we regard topic maps as complex networks, the number of associations referred by a topic should obey a power law, not a normal distribution. Since the distribution is too skewed, this may indicate that we should not adopt the mean degree as the typical number of associations.

Other factors such as variation in the size of objects may also affect the precision. This is because the effect of small errors can be large by multiplying a large number such as the number of topics or associations.

Taking these factors into account may improve the precision of our formula.

## 5 Conclusion

In this paper, we introduced the execution plan optimization method for queries submitted to topic map databases. We assumed tolog-type query language and the topic map data structure defined by TMDM.

In order to choose a suitable plan for processing queries, we introduced the estimation formula for the retrieval cost of each plan, which includes the cost of objects loaded from a storage device into memory. Among query syntaxes defined by tolog, we focus on the syntax that supports the query introduced in the last section. Our target query is to search topic objects referred by a specific association with a particular topic specified in the query.

Applying our formula to the Rampo Edogawa topic map and the Pokemon topic map, we identified the retrieval cost of objects that is unique to the class of information items (such as topics, topic names, associations, and association roles) and compared the execution time needed for the three queries for each topic map with the tendency of the value of cost. Both for the execution time and for the estimated cost, we saw the tendency that it is effective to execute the association route for the Rampo Edogawa topic map and the topic route for the Pokemon topic map, as far as the order of magnitude of the

ratio is concerned. This coincides with our intuition, since fewer associations than topics are included in the Rampo Edogawa and fewer topics than associations are included in the Pokemon topic map.

The estimated value of our formula has a similar order of magnitude to the measured execution time. The reason for the errors may originate from simplification in assuming the distribution of data. Although we assumed that the number of associations referred by a topic is uniform for simplicity, the number of associations should obey a power law if we regard topic maps as complex networks.

In the future, we will improve the precision of our estimation method, as well as improving the formula. We will also apply our method to the other query syntaxes that have plural routes to access objects as shown in 2.3.

## References

1. Ontopia, tolog Language tutorial, http://www.ontopia.net/
2. ISO/IEC JTC1/SC34, Topic Map – Data Model, http://www.isotopicmaps.org/sam/sam-model/
3. Yuuki Kuribara, Takeshi Hosoya, and Masaomi Kimura: TOME: Topic Maps Database Extended
4. db4o, http://www.db4o.com/
5. Pokemon Topic Map, http://www.ontopia.net/omnigator/models/topicmap_complete.jsp?tm=pokemon.ltm

# Part III

# Information Wants to be a Topic Map

# Topic Maps for Improved Access to and Use of Content in Relational Databases – a Case Study on the Descriptive Variety Lists of Germany's Bundessortenamt

Gerhard E. Weber, Ralf Eilbracht, and Stefan Kesberg

Nexxor GmbH, Vollmoellerstr. 11,
70563 Stuttgart, Germany
\{gerhard.weber,ralf.eilbracht,stefan.kesberg\}@nexxor.de

**Abstract.** Information delivery via web-based access to databases is ubiquitous. Nevertheless, the wealth of information held in many of these resources is poorly accessible, due also to the limited number of views provided, and frontends merely reflecting their backends' data-centric relational information architectures. We suggest using Topic Maps-based web frontends on top of relational resources for improving their usability by subject-centric information delivery, and demonstrate the approach with the use case of the descriptive variety lists of Germany's Bundessortenamt.

## 1 Introduction

The entity relationship model and a data-centric publishing paradigm still dominate the logic behind information resources. The ER-model is indeed an efficient solution for creating vast data stores, which may be accessed with sophisticated SQL-queries. However, most users cannot code SQL-queries, and are also not prone to efficiently use tools supporting the creation of queries appropriate for their purposes. Furthermore, the number of possibly relevant views on any database of some complexity is too large to be met by preconfigured user interfaces. Rather than restricting the usability of resources by providing only a limited number of relational views, we advocate offering associative subject-centric access paths along whatever axis of interest users would want to pursue, in combination with automatically generated relational views rendered from the graph structure of networked subjects. To that end, we suggest using Topic Maps-based web interfaces on top of relational resources for improving accessibility and usability of content, and demonstrate the approach with the use case of the descriptive variety lists of Germany's Bundessortenamt.

As an independent federal authority under the supervision of Germany's Federal Ministry of Food, Agriculture and Consumer Protection, the Bundessortenamt is responsible for granting of plant breeders' rights, and registering of varieties in the descriptive variety lists, which is a legal precondition for commercializing seed of agricultural species and vegetables. It caters to consumer protection and ensures high quality seed and planting stock material of resistant and high performance varieties for farmers and horticulturists.

Addition to the descriptive variety lists requires successful performance of a number of multi-site tests over several years, including characterization of properties such as cultivation, resistance, yield, internal quality and processing characteristics. The Bundessortenamt publishes the results of these tests for the registered varieties. Thus, the descriptive variety lists describe the value features of varieties, and are used by farmers, growers, advisors, co-operatives and the processing industry. The authority publishes the lists in print, as downloadable PDF-files, and also offers a web frontend on a relational data base.

The major aim for the use-case example was subject-centric, straightforward access to views required for answering relevant questions on the subject domain, such as quick identification of variety distributions over the spectrum of property values in support of the assessment of particular varieties. Also, questions related to breeder activity were to be addressed. Increasing content usability also required the provision of an efficient tool set for sorting, and filtering of relational views.

We briefly describe the database web frontend provided by the Bundessortenamt, and highlight its limitations with a number of relevant questions. Based on the publicly available database, we developed an ontology, and transformed all content into a topic map. The web application rendering access to the topic map is based on a commercial software package for processing topic maps.

## 2 Original Data and Web Application

Data published by the Bundessortenamt describes 88 assortments, some of which are described with more than 30 kinds of properties. More than 200 breeders registered over 2600 varieties. The web interface of the authority's database provides an overview of the assortments, and one to two tables for each assortment listing all registered varieties and their property values mostly coded in a numerical ranking scale.

Whereas the web interface does provide access to the large number of the varieties' evaluated features, the very low number of views and the data-centric content provision restrict the utility for answering highly relevant questions for experts in the farming, extension, plant breeding or crop processing domains.

Some examples for questions hard or impossible to answer with the authority's web application are listed in Table 1. Gaining quick access to the distributions mentioned in questions No. 1 and 2, would greatly enhance the users' ability to judge a particular variety with respect to the total population of an assortment. Furthermore, these distributions would help to identify and aid the assessment of potential breeding targets. Due to disease impact on yield, and the cost of disease control, disease susceptibility is a critical kind of property for many crops, which puts questions No. 3 and 4 into the focus of interested parties. Questions No. 5 and 6 are related to business intelligence issues for plant breeders but may also be relevant for decisions on public funding for breeding projects.

The failure to answer these particular questions with the authority's web application is not due to limitations of its underlying relational data model, but to the restricted effort invested in the web frontend. By configuring the appropriate views and providing some functionality for handling data in tables, it would be possible to retrieve the answers

**Table 1.** Some questions hard or impossible to answer with the web application of the descriptive variety lists of the Bundessortenamt

| No. | Question |
|---|---|
| 1 | What is the age distribution of the registered varieties in a particular assortment? |
| 2 | What is the variety distribution over a particular kind of property in a particular assortment, and how is a particular variety positioned in this distribution? |
| 3 | Which variety of a particular assortment shows the lowest susceptibility to a particular disease? |
| 4 | Which variety of a particular assortment and breeder shows the lowest susceptibility to a particular disease? |
| 5 | What are the assortments of activity of a particular breeder? |
| 6 | Which breeders dominate in a particular assortment, and to what extent? |

to the listed questions. However, users may want to pose a multitude of additional questions, and for a large share of them, appropriate SQL-queries are required for finding the answers. Therefore, the relational data model in combination with limited resources for providing appropriate views are indeed restricting the usability of valuable content.

## 3 Topic Map Creation

Original data was downloaded as ASCII and html-files from the authority's website. Subsequently to the analysis of content and data structures, a Topic Maps ontology was created, relying on inhouse expertise in the application domain. Effort for ontology design was kept at a minimum, and the requirement of a realistic ontology advocated by Weber et al. [1] was not adhered to strictly. Data-driven constructs were accepted, where they brought a clear benefit in terms of reduced cost for frontend development.

Data preprocessing was a prerequisite for subject-centric mappings. In particular, address fields had to be preprocessed to enable entity recognition of locations, breeders, and breeder types such as person, company, and organization.

For entity recognition of non-German headquarter locations of breeders, we restricted details to the national level, whereas for German locations we identified the community level. Location data provided by the authority was supplemented with additional information on hierarchical structures such as county, or higher administrative structures. For the sake of providing a single root subject for locations, we also registered all relevant continents as parts of the world.

For improved access we introduced specialization-relationships for assortments and kinds of properties. Hierarchies were defined in separate topic maps and subsequently merged.

Based on the application ontology and the structure of the data sources, mapping rules were defined for transforming all content of the variety lists into topic maps [2]. Assortments, varieties, breeders, kinds of properties, property values expressed as ranks or calendar years, and headquarter locations of breeders were mapped as topics. Consequently, most property values of the varieties were modeled by associations with

their respective variety. Attributes such as breeder address, total plantation area for seed production, and registration numbers of plant varieties were modeled as occurrences.

The resulting topic map, coded in XTM 2.0 [3], contains some 1600 topic types, nine association types, 16 role types and 13 occurrence types. The total number of topics, associations and occurrences is approximately 20.000, 80.000, and 16.000 respectively.

## 4 Topic Map-based Web Application

We used topicWorks, a commercial software package, for processing the topic map, and for creating the publicly available web application. The frontend is based on a generic Topic Maps browser rendering most interface structures from the processed topic map. A template system and configurable interface structures enable customized views for constructs of particular interest. Templates and view customization were used for ontology topics such as assortment, variety, breeder, or location.

The generic browser distinguishes two major visualization patterns. For topic types, tables of instances and their statements are generated by default. For non-typing topics, e.g. a particular variety, or a particular breeder, individual views are generated, collocating all statements on the focal topic. Templates and configuration allow for the modification of the default visualization patterns, and any required view may be created. Due to the two default patterns, the generic Topic Maps browser constitutes a web application out-of-the-box, and renders useful web pages even without any configuration at all. However, configured views based on tolog [4] queries increase the application's utility.

Figure 1. shows a screen shot of the overview on winter wheat varieties, addressing question No. 4 in Table 1. The view displayed is one click away from the assortment's overview, and requires entering three characters for text filtering and another click for sorting the critical column. The 143 varieties of this assortment are filtered for the breeder of interest, by entering "kws" in the text filter box to the left above the table header. Subsequently, sorting the susceptibility ratings for mildew (Mehltau) in ascending order, retrieves "Dekan" as the least susceptible variety for this disease of interest. Answering this question on the basis of the authority's web application would require hours of tedious research, and is thus practically not possible.

In contrast to the low number of data-centric relational views provided by the authority's web application, the Topic Maps-based application renders several thousand subject-centric relational views, many of which are not customized but generated automatically due to the frontend's generic visualization pattern. Thus, neither do developers need to code and configure a large number of SQL-queries at design time, nor do users have to code tolog-queries. Any of these automatically generated views can answer questions, no developer would have had to anticipate at design time.

Figure 2 shows such an automatically generated relational view addressing question No. 2. The distribution of winter barley varieties over the range of susceptibilities to a particular disease is displayed. The highlighted "Wendy" is revealed as one of the assortment's least susceptible varieties to the critical disease. Whereas, with the authority's web application, this question is not answerable, with the Topic Maps-based application, the required view can be retrieved at the cost of a click in the respective column header of the assortment's overview.

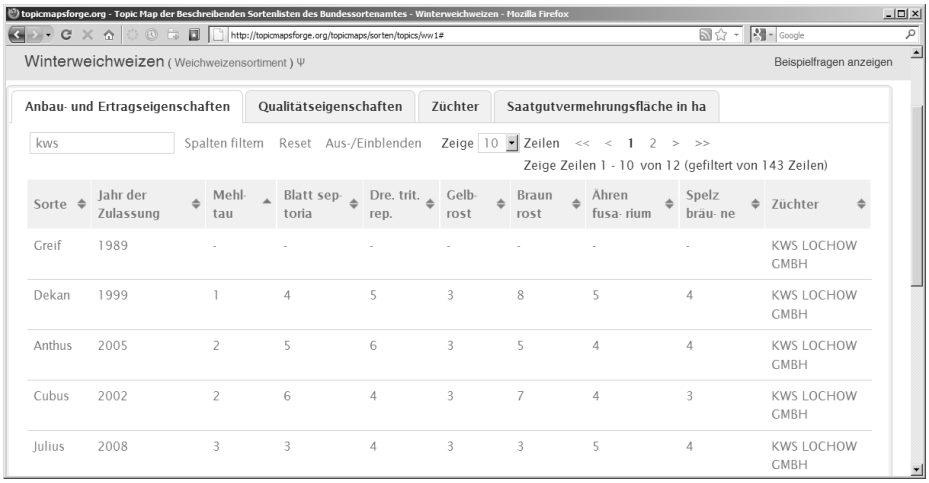



**Fig. 1.** Screenshot of the Topic Maps web frontend displaying a selection of properties of winter wheat varieties filtered for a particular breeder and ordered by ascending susceptibility for a particular disease; note that all content in the table and the column headers is provided not just as strings or characters but as topics, each of which offers access to related information

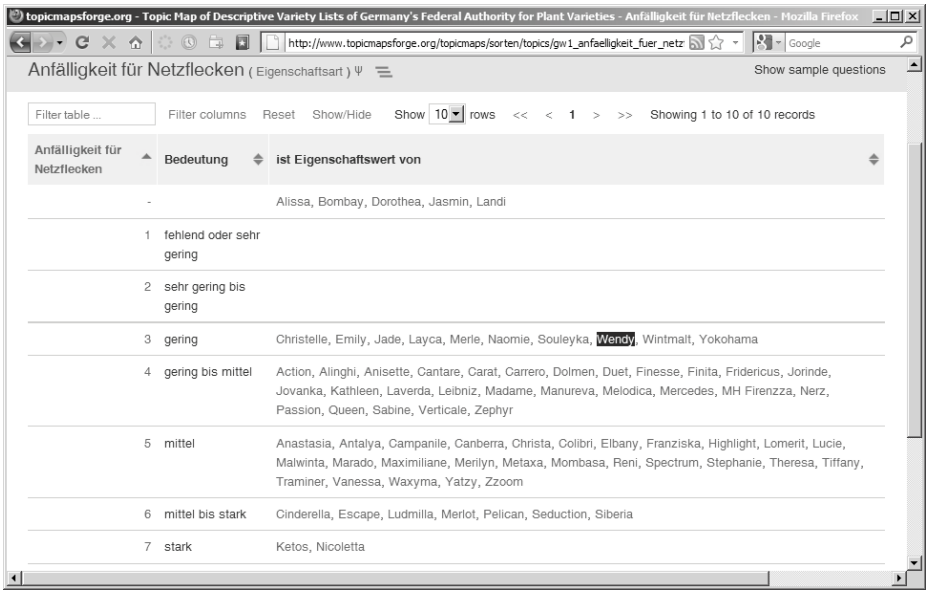



**Fig. 2.** Screenshot of the Topic Maps web frontend showing the distribution of winter barley varieties over the susceptibility range for a particular barley disease

The view displayed in Fig. 2 may also be retrieved by a number of other access paths. Since most property values are modeled as topics, all varieties with a particular property value may be retrieved by a click on this very value. In the next step, the type of this value – displayed with the topic name in the header area – offers access to relational views listing all property values of the respective type, and the assigned varieties.

We did not perform a systematic survey of user experiences with the Topic Maps-based web frontend. However, we asked eight agricultural experts for informal feedback, after supplying the web application's URL. All of them appreciated the ease of data retrieval. One of them criticized the lack of faceted search, which actually is available for each relational view, but obviously not immediately recognizable to all users. As a consequence, we implemented an online manual for the platform we used for web publishing the topic map.

## 5 Discussion

By using a domain ontology and a mapping process subsequent to some data preprocessing, we created a topic map from the relational database for properties of plant varieties registered in Germany. We supplemented content retrieved from the authority's database by containment-relationships for geopolitical units. In addition, we also supplemented specialization-hierarchies for assortments, and kinds of properties, defined on the basis of our own expertise in the agricultural domain. Based on commercial Topic Maps software, we provided a subject-centric web application, some views of which were configured with a number of tolog queries including inference rules where appropriate.

There are a number of options for providing subject-centric access to content in data-centric relational data stores. Next to complete mappings of sources, on-demand mapping by SQL-generation and subsequent transformation of the result sets may be employed. We decided for a complete mapping for a number of reasons. Firstly, we did not have direct access to the relational database, but used downloaded CSV- and HTML-files. Secondly, data of the variety lists shows little dynamics and is updated annually only. Thirdly, for the intended showcase, we did not need to provide a productive system with automated updating, but considered a static mapping sufficient to make our point. Fourthly, by providing topic map content held in memory, we reduce latency due to network performance, SQL processing time of the authority's server, and processing time for transformation of the result sets. Thus, whereas automated real-time subject-centric access to relational data stores is possible, we emphasize that this was not an issue for our use case.

We demonstrated how a Topic Maps-based web application on top of a relational data store can improve accessibility and usability of content. The authority's web interface provides a low number of data-oriented views, and does not support answering a host of relevant questions on its domain. In contrast, the subject-centric web interface based on the topic map transformation offers a densely knitted, navigable network of associated subjects, and provides a large number of overviews in support of swift answers. Contrary to an application based on a data-centric relational model, a subject-centric Topic Maps-based frontend does not restrict users to views preconfigured at design time. The Topic Maps-frontend automatically renders relational views from its underlying graph structure.

Thus, even without any configuration of frontend structures, hundreds of networked relational views become available as opposed to the small numbers of views usually provided for data-centric relational databases.

The benefits of Topic Maps-based frontends for accessing scientific data have been demonstrated by Weber et al. [5] for the results of a European ring test on the ecotoxicity of waste, as well as by Husaková and Olševicová [6] for an astronomical use case. Going beyond single applications, Topic Maps-based integration of heterogeneous data sources allows solutions scaling to the web, as Stümpflen et al. [7] showed, even for very large volumes of life science data. For data sources with subject-centric web interfaces, providing respective web-services is the obvious next step. To remain in the domain of our use case, subject-centric information retrieval could thus seamlessly integrate resources such as the descriptive variety lists with data on agrochemicals for crop protection published by Germany's Federal Office of Consumer Protection and Food Safety.

Understandably, a large amount of effort goes into acquiring a wealth of scientific data on the varieties which are the basis of Germany's crop production. Less understandably, the efforts for making these data available, and – to name it more precisely – for transforming them into readily available knowledge, fall short of what is possible today. This imbalance between big efforts allotted to data-acquisition and the largely cost-minimizing regimes directing data curation and publication seems common in many domains. All the more, subject-centric information architecture, and Topic Maps-based web publication of scientific data should be considered as a pragmatic solution for an omnipresent problem.

## 6 Conclusion

Topic Maps enables user interfaces for better use of content in relational databases by upgrading content from data to networked knowledge models with multitudes of subject-centric access paths, and large numbers of relational views automatically rendered from the underlying graph structures.

## Acknowledgements

We thank the three anonymous reviewers, whose comments and suggestions helped improving this manuscript.

## References

1. Weber GE, Eilbracht R, Kesberg S, 2008. Topic Maps as application data model for subject-centric applications. In: Maicher L, Garshol LM (eds.) Proceedings of TMRA 2008, Fourth International Conference on Topic Maps Research and Applications, Leipzig, Germany, 15–17 October 2008. Leipziger Beiträge zur Informatik XII, 1–9
2. ISO/IEC 13250-2: Information Technology – Topic Maps – Part 2: Data Model. International Organization for Standardization, 2006

3. ISO/IEC 13250-3: Information Technology – Topic Maps – Part 3: XML Syntax. International Organization for Standardization, 2007
4. Garshol LM, 2006. tolog – A Topic Maps Query Language. Maicher L, Park J (eds.) Charting the Topic Maps Research and Applications Landscape – First International Workshop on Topic Map Research and Applications, TMRA 2005, Leipzig, Germany, October 6–7, 2005. Springer, Berlin, 183–196
5. Weber GE, Eilbracht R, Kesberg S, 2009. H14-Navigator uses Topic Maps as application data model. In: Römbke J, Moser H (eds.) Ecotoxicological characterisation of waste – Results and experiences from an European ring test. Springer, New York, 259–270
6. Husáková M, Olševicová K, 2008. Creating Web Presentation for Observatory and Planetarium with Topic Maps. Maicher L, Garshol LM (eds.) Proceedings of TMRA 2008, Fourth International Conference on Topic Maps Research and Applications, Leipzig, Germany, 15–17 October 2008. Leipziger Beiträge zur Informatik XII, 237–246
7. Stümpflen V, Nenova K, Barnickel T, 2008. Large Scale Knowledge Representation of Distributed Biomedical Information. Maicher L, Garshol LM (eds.) Proceedings of TMRA 2007, Third International Conference on Topic Maps Research and Applications, Leipzig, Germany, 11–12 October 2007. Springer, Berlin, 116–127

# Part IV

# Optimizing Data Access

# Spatial Identification of Subjects

Sven Krosse

University of Leipzig, Johannisgasse 26, 04103 Leipzig, Germany
krosse@informatik.uni-leipzig.de

**Abstract.** Modeling information resources by using the Topic Maps paradigm draws its benefit from the fact that each real world subject has to be represented as exactly one resource item in the data model. Thus, Topic Maps stores have to guarantee, that two information items representing the same subject, will be merged automatically. Because of that the model should never contain two identical subjects. The main problem is the weak identity in current Topic Maps engines, which is often defined by humans and not by domain-specific algorithms or patterns. The special domain of spatial (geographical) knowledge provides such an algorithm based on a standardized geographical system, like WGS-84. In this case, a Topic Maps engine can provide a powerful algorithm for unique identities of subjects and, moreover, additional knowledge about the subjec can be drawn from the geographical information.

## 1 Introduction

One of the main benefits of the Topic Maps meta model became one of its main problems – the automatic merging process. The power of merging algorithms depends on the quality of subject identities. Two topics can only be identified as equal, if both information items use the same identity, but the model does not contain any restrictions for the used identities themselves which often results in identities being weak. The usage of domain-specific knowledge can help to overcome this flaw. For instance, in the domain of spatial (geographical) information, the knowledge of the domain can be used to generate subject-locators, which are unique all over the world.

The problem of identity weakness is caused by the lack of ability to represent real domain knowledge, using the standardized Topic Maps API (TMAPI) [1]. Information resources are simply taken to be 2-dimensional constructs, without any domain-specific knowledge. But for using domain-specific algorithms, it is necessary to represent such information at the abstract upper layer of TMAPI.

An alternative might be the definition of standardized workflows for handling domain-specific identities in Topic Maps engines. As the goal of the Topic Maps Data Model (TMDM) is to form a technology for encoding knowledge and to connect it with knowledge-relevant information [5], it is necessary to use the whole domain-specific knowledge, e.g. the spatial dimension. Note that the spatial domain is only one example that provides a powerful identification algorithm for subjects of the real world.

## 2 Identity Dependency of the Merging Process

The main benefit of the Topic Maps paradigm is the ability to represent each real world subject as a corresponding unique topic. Subjects representing the same real world item can be detected automatically, based on a set of rules and should be merged by the underlying Topic Maps backend.

The TMDM [5] defines a set of merging rules to identify topics representing the same subject. Most of them are based on identity constraints of topic items, which narrows the concept of identity, such that two topic items are the same if the use the same identity. The main problem is the identity itself. The identity of a topic item is defined to be a set of subject-identifiers and subject-locators both of which are usually manually defined.

In this case two information items are identified as equal if and only if two people or algorithms use the same identifier for both subjects. The problem of this weak identification can be clarified by a simple scenario. If there are two people, both creating a statement about the city of *Leipzig* and they decide to use the URI of the *wikipedia* page, but in different language scopes, the merged topic map will contain two topics for the same subject, as the following CTM snippet shows.

```
%encoding "UTF-8"
%version 1.0
<http://de.wikipedia.org/wiki/Leipzig>.
<http://en.wikipedia.org/wiki/Leipzig>.
```

Another problem can be the ambiguity of name-based IRI representations as subject identities. The IRI *http://psi.example.org/city/Leipzig* may be used as a subject-identifier of a city called *Leipzig*. But this could be a city located in Canada or in Germany. The URI literal cannot differentiate between different locations.

In general, we have to decide, whether it is possible to use the unique global geographical identity of a city as a unique identification statement or if there are possible solutions for weak identities, like Subj3ct [3].

### 2.1 The Subject Identity Resolution Service – Subj3ct

The online resolution service Subj3ct [3] tries to resolve the identity problem of subjects by providing a repository for subject-identifiers. An application can use the online service to extract the identity of a subject about which it wants to make a statement. The web service is organized as powerful registry which can be extended by external applications. Therefore, applications have to provide new identities of an existing or a new subject. The new identity will then be evaluated by a complex trust system. The trust algorithm calculates the trust level of the reporting application depending on the used IRI namespace. The score of subject-identifiers located in the namespace of the reported application will be much higher than outside of the IRI namespaces.

However, the identity problem cannot be solved completely by the Subj3ct webservice. There are no subject-identifiers for subjects, the registry does not know at query time. If there is insufficient knowledge to build a query or if the query is ambiguous, the quality of the received identifiers appears to be problematical. The problem of ambiguity is constituted by the fact that the query is based on known subject-identifiers or names of

the requested topic. Whenever an application only has few information about a subject the result can be empty or ambiguous. In this case, we need a powerful algorithm or standardized patterns to generate unique subject-identities based on a small knowledge base.

The design of Subj3ct being a webservice can become a problem, too, because the application needs web access all the time. An offline version of Subj3ct, on the other hand, will create an isolated world and would result in different subject-identifiers too.

However, the online resolution service is one powerful way to retrieve identifiers for a specific subject with the downside of only being able in a web environment. As shown, it cannot satisfy all requirements for a unique identity solution, however, the webservice can be used to store such standardized locators of the spatial domain.

### 2.2 Spatial Information as Identity

The identity problem of subjects can be considered similar to the identification problem of geographical locations, like cities or countries on the surface of the earth. Each city, country or other location on earth has to be identified by each geographical information system in a common way, so that equal positions on earth are equally identified. This paradigm is fundamental for routing algorithms and global navigation systems (GNS).

Global navigation systems have to use the same data exchange formats to allow communication between different information providers. The definition of point of interests (POI) is only one example to clarify the importance of a standardized system for geographical coordinates. Because of that, an international system for global position identification was established. The World Geodetic System (WGS-84) [4] is used by most information systems working on top of geographical knowledge. Based on this standardized system, different applications and data providers can work on the same data sets and can identify spatial locations in a common way.

Topic Maps are containing spatial information too, for instance countries or birth-places of persons. In this special domain the identity problem can be solved by using domain-specific knowledge through generating subject-locators based on the WGS-84 system.

## 3 Spatial Subject-Locators

If a subject represents a geographical location, like a city, a country or a point, it can be addressed by a geographical coordinate using one of the defined coordinate reference systems (CRS), like WGS-84. To realize an engine-independent representation of such identities, it is necessary to use existing standards or patterns of other domains, addressing and solving the same problem [5,6].

The RFC 5870 tries to define a standardized representation of physical locations in a two- or three-dimensional CRS. The reference to such a physical location is similar to the concept of subject-locators, which should be a reference to an information resource too. Because of that, the spatial locators can be defined as subject-locators referencing physical resources.

### 3.1 Uniform Resource Identifier (URI) for Geographical Locations

The main problem addressed by the RFC 5870 [5] is the way of representing spatial locations in different data models and structures. This representations are independent from each other and disables any interoperability, because of that a standardized URI scheme was designed.

The 'geo' URI scheme identifies a location in a specific reference system, like the WGS-84, which is also used as default CRS. Each location can be represented in two- or three dimensions (latitude, longitude and optionally altitude).

**World Geodetic System 1984 – WGS-84** The World Geodetic System was established in the year 1984 to create a worldwide valid geographical identification method. The model defines a special ellipsoid to normalize the form of the earth body in the best way.

Two-dimensional coordinates are represented as a comma-seperated pair of latitude and longitude values, measured in decimal degrees. The optional third dimension represents the altitude value, measured in meters. The latitude values range from -90 to 90 degrees (reflecting the poles) and the longitude values range from -180 to 180 degrees. The negative values represent coordinates in the Southern and Western hemispheres or altitudes below the geoid's surface.

Sometimes WGS-84 coordinates are specified as sexagesimal[1] values. Both descriptions can be transformed into each other (1,2).

$$\text{dcdegree} = \frac{\frac{\text{seconds}}{60} + \text{minute}}{60} + \text{degree} \tag{1}$$

$$\begin{aligned} \text{degree} &= \lfloor \text{dcdegree} \rfloor \\ \text{minute} &= \lfloor (\text{dcdegree} - \text{degree}) \times 60 \rfloor \\ \text{second} &= \lfloor (\text{dcdegree} - \text{degree}) \times 60 - \text{minute} \rfloor \end{aligned} \tag{2}$$

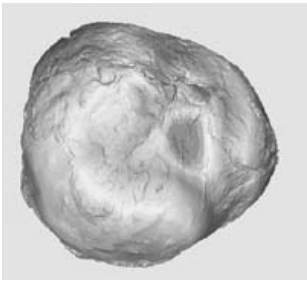


**Fig. 1.** Abstract model of the WGS-84 geoid [7]

[1] A numerical system with sixty as its base.

**URI Scheme Syntax** The 'geo' URI defines a scheme syntax of such special geographical URIs. An section of the scheme syntax as Backus-Naur Form (ABNF) are included below.

```
geo-URI     = geo-scheme ':' geo-path
geo-scheme  = 'geo'
geo-path    = coordinates p
coordinates = coord-a ',' coord-b [ ',' coord-c ]
p           = [ crsp ] [uncp] *parameter
crsp        = ';crs=' crslabel
crslabel    = 'wgs84' | labeltext
uncp        = ';u=' pnum
```

The meaning of the three coordinate values (*coord-a*, *coord-b* and *coord-c*) depends on the used optional CRS parameter part defined by the *crsp* part of the URI. If the CRS is unspecified, the default WGS-84 should be used. In this case, the latitude will be represented by the *coord-a* argument, the longitude by *coord-b* and the optional altitude by *coord-c*. All values are measured as decimal degrees or meters (altitude). The number of digits of the coordinate values may not be interpreted as a level of uncertainty. The uncertainty will be defined as a value in meters by the second parameter part *uncp*. The application or the protocol has to decide about its interpretation.

### 3.2 Geo URI as Subject-Locators

The standardized URI's for geographical locations can be used as a domain specific solution for the identity problem of Topic Maps models. In the spatial domain the 'geo' URI represents a subject-locator for a subject, representing a physical geographical resource. These subject-locators can be interpreted by every Topic Maps store, independent of the programming language or modeled ontology.

### 3.3 Equality Detection

The domain-specific subject-locators often need special equality detection routines, being more complicated than a simple string comparison. The RFC 5870 [5] defines the equality requirement of two 'geo' URIs as a set of conditions.

Two 'geo' URIs are equal if and only if they use the same reference system defined by the *crs* parameter. The default system and the parameter argument *wgs84* are equal by definition. The coordinates and uncertainty values of both URIs have to be mathematically equal. The values cannot be compared based on their string representation because the same number might be represented as different string literals depending on the number of digits.

Additionally, each CRS can define a number of equality constraints that should also be handled by the engine. For instance the WGS-84 CRS defines, that the values 0 and -0 are equal and the longitude value must be ignored if the latitude is set to either 90 or -90 degrees. The longitude values 180 and -180 are considered as equal too, because of the *dataline case*.

**Comparison Examples** This short section contains an examples for the ‘geo’ URI comparison, which are extracted from the RFC 5870 document [5].

There are two subjects with the subject-locators <geo:90,46> and <geo:90,-22.43; crs=WGS84>. Both subjects are equal because they use the WGS-84 CRS and reflect a location at the north pole. In this special case the model dictates that the longitudes have to be ignored.

### 3.4 Problems

The standardized spatial subject-locators can be used to identify subjects representing geographical locations in the real world. Thus, their coordinates have to be static, but the geographical coordinate of a complex object like a city is often simply a point, representing the center of this city. Over the time, the city can change its dimensions which also involves the movement of the city center. The subject-locator of such a city would be changed too. Because of that, Topic Maps engines cannot identify two topics as equal, representing the city at different times.

The engines have to adjust temporal changes by using a special identification algorithm for such spatial subject-locators supporting uncertainty. Therefore, the equality of subject-locators should not be based on the exact equivalence of their coordinates’ literals, but on the maximum deviance between their geographical identities. The value of maximum deviance or uncertainty cannot be defined in a common way, because it depends on the level of detail that the ontology describes. For instance if the topic map wants to make statements about the districts of a city, instead of the whole city, the deviance value should be much smaller than for the whole city or, more precise, smaller than the maximal dimension of the objects contained by the ontology.

**Identity Detection Based on GeoNames API** A special implementation of such an identification algorithm for adjusting spatial locality over the time can be based on the GeoNames [8] web services. The most important assumption to use the web service of GeoNames is the locality of geographical changes, which means that the spatial identity of a geographical subject is changed without changing its neighborhood. Therefore, the distance between the old and the new location has to be smaller than to any other location.

If this requirment is met by the time model, the web service method *findNearby* can be used by the Topic Maps engine. This method requires a 2-dimensional WGS-84 coordinate (*lat* and *lng*) and returns the location with the smallest distance to the defined coordinates. Therefore, both arguments can be extracted from the spatial subject-locator in case the WGS-84 CRS is used. The result will be an XML, RDF or JSON fragment, containing the information about the location found nearby the given coordinates. The following XML fragment displays the result of the query
http://ws.geonames.org/findNearby?lat=51.339444&lng=12.371111.

```
<geonames>
   <geoname>
     <toponymName>Leipzig</toponymName>
     <name>Leipzig</name>
     <lat>51.33962</lat>
```

```
      <lng>12.371292</lng>
      <geonameId>2879139</geonameId>
      <countryCode>DE</countryCode>
      <countryName>Germany</countryName>
      <fcl>P</fcl>
      <fcode>PPL</fcode>
      <distance>0.0233</distance>
   </geoname>
</geonames>
```

The Topic Maps engine can then use the provided latitude and longitude values, to generate a second subject-locator as 'geo' URI for the topic and merge both topics representing the same city.

**Using Complex Geographical Shapes** Another possibility to solve this problem, is the shape-based representation of such geographical locations using a standardized syntax, like the Geography Markup Language (GML) [9]. This solution is not considered inside the scope of this document and thus is not discussed.

## 4 Additional Spatial Information of Subjects

Additionally, the domain knowledge can be used for extracting dependent information of geographical subjects, like relationships to other subjects of the topic map.

### 4.1 The GeoNames Webservice

The GeoNames [8] web services are mainly REST webservices to provide information about a geographical subject. Therefore, the subject will be identified by its latitude and longitude values, one of its names or, specitically its GeoNames id. The web service contains a set of methods for extracting additional knowledge provided by the spatial domain.

All information will be returned as a XML, RDF or JSON resource which can be parsed by the client application to extract data resources and additional domain-specific knowledge.

**Table 1.** Some of GeoNames webservice methods [8]

| Method name | Attributes | Description |
|---|---|---|
| findNearby | latitude (*lat*), longitude (*lng*) | Returning the geographical location nearby the given geographical point. |
| extendedFindNearby | latitude (*lat*), longitude (*lng*) | Returning the geographical hierarchy nearby the given geographical point. |
| neighbours | GeoNames id (*geonameId*) | Returning all neighbouring countries of the given country. |

**Table 2.** Standardized identifiers for spatial associations

| subject-identifier | description |
|---|---|
| spatial:containment | the containment of spatial subjects |
| spatial:container | the role type of a containment |
| spatial:containee | the other role type of a containment |
| spatial:neighbourhood | the spatial neighbourhood of subjects |
| spatial:neighbour | the role type of spatial neighbourhood |

### 4.2 GeoNames-Provided Domain Knowledge

As a result of unique identification of geographical subjects by their spatial subject-locators, the engine can provide additional information using the web services of GeoNames [8].

For instance the *findNearby* method provides the possibility to extract default names of subjects by their WGS-84 coordinates. The returned XML file contains the tags *name* and *toponymName* which can be used.

Normally, the user has to design all relations between geographical subjects, like countries and their cities if the topic map should represent the spatial domain at a specific level of detail. In this case the information are modeled twice, because the geographical domain and the coordinates of the subjects implicate these relationships, too. By using the webservice *extendedFindNearby* the engine can create the geographical hierarchy automatically, so that, the associations can be generated with types reflecting all geographical linkage. Thus, these topic types should use standardized subject-identifiers in the namespace '*http://psi.topicmaps.org/iso13250/glossary/spatial/*' (represented by the prefix *spatial*). For instance the containment of a city in a country can be represented by the topic type *spatial:containment*.

Table 2 contains a set of such standardized subject-identifiers for spatial linkage.

## 5 Summary

Using a standardized geographical system, like the described WGS-84 system, subjects representing a location (city, country etc.) of the real world, can be identified in an automatic and standardized way to solve the identity problem in this special domain. The geographical identification can be transformed into the Topic Maps paradigm by generating subject-locators as 'geo' URIs based on this spatial knowledge. A secondary effect of this unique identification algorithm is the possibility of using existing geographic information system (GIS) or webservices, like GeoNames, to extract additional knowledge contained in the current domain. A Topic Maps engine can be more powerful than current implementations. The benefits are:

- the automatic generation of unique identities
- the powerful and standardized detection of identical subjects by using their geographical identity (not only for Topic Maps)
- the extraction of default names and other knowledge
- the generation of spatial association items

However, the identity problem cannot be solved in a common way, but the domain-specific knowledge provides good possibilities for some solutions as shown for the spatial (geographical) domain.

## References

1. Heuer, L., Schmidt, J.: TMAPI 2.0. In: Maicher, L.; Garshol, L. M.: Subject-centric Computing. pp. 129–136, Springer, Berlin (2008)
2. ISO/IEC IS 13250-2: Information Technology – Document Description and Processing Languages – Topic Maps – Data Model. International Organization for Standardization, Geneva, Switzerland, 2008-06-03
http://www.isotopicmaps.org/sam/sam-model/
3. Moore, G., Ahmed, K.: Subj3ct – A Subject Identity Resolution Service In: Maicher, L.; Garshol, L. M.: Linked Topic Maps pp. 163–172 Springer, Berlin (2009)
4. National Imagery and Mapping Agency: Department of Defence World Geodetic System 1984, Third Edition, Technical Report 8350.2, 2000-01-03
http://earth-info.nga.mil/GandG/publications/tr8350.2/WGS84fin.pdf
5. Internet Engineering Task Force (IETF): A Uniform Resoure Identifier for Geographic Locations ('geo' URI), RFC 5870, June 2010
6. International Organization of Standardization: Standard representation of geographic point location by coordinates, ISO Standard 6709, 2008
7. Helmholtz-Zentrum Potsdam Deutsches Geo-Forschungs-Zentrum http://www.gfz-potsdam.de/portal/gfz/home
8. GeoNames A community driven geographical database http://www.geonames.org/
9. International Organization of Standardization: Geographic information – Geography Markup Language (GML), ISO Standard 19136, 2007

# Defining Domain-Specific Facets for Topic Maps with TMQL Path Expressions

Sven Windisch and Lutz Maicher

University of Leipzig, Johannisgasse 26, 04103 Leipzig, Germany
\{windisch,maicher\}@informatik.uni-leipzig.de

**Abstract.** The automatic generation of facets works fairly bad for fine-modeled ontologies, in which not all information concerning a single Topic is available through occurrences and direct associations. In this paper, we share our conception of using TMQL path expressions for the definition of domain-specific facets by means of using standard-based Topic Maps technologies. The generated facets must be evaluated, even though they are defined manually by a domain expert. We therefore propose metrics for automatic evaluation of the defined facets, as well as a mechanism for using automatically stored user feedback.

## 1 Introduction

Traditional keyword-driven search adheres the *informational search* paradigm, where a user searches for an item of which he has no knowledge, but expects to find in the given full-text index by providing the correct keywords. After retrieving some search results, the user then employs these results as starting points for *navigational search*. That means, he has knowledge about some items of the data base and wants to investigate related items to gain more knowledge [1]. Moreover, the user may be overwhelmed by an exceedingly large number of search results. In this case, he may want to apply constraints to narrow the number of results. The use of facets for search results is a solution to both the need for navigating through the data base and narrowing down a flood of search results.

According to Taylor, a facet embodies "clearly defined, mutually exclusive, and collectively exhaustive aspects, properties or characteristics" of a specific subject [2]. For example, a person may have a name facet, a birthday facet, a residence facet, etc. Following the Topic Maps paradigm, each subject is represented by a Topic, which by itself carries the characteristics of the subject in constructs like names, occurrences and associations and all of them can be transformed into generic facets, as was shown by Ueberall and Drobnik [3]. They propose the use of associations and occurrences as candidates for the automatic creation of Topic-centric facets.

This approach performs well for coarse-modeled ontologies, in which Topics have all their important characteristics stored in occurrences or *direct associations*. Direct associations are $n$-ary associations whose players are connected directly without any interstation (e.g., the person-address-association in Fig. 1). Unfortunately, this approach does not work very well for fine-grained ontologies, where not all information concerning a single Topic is available through direct associations (e.g., to retrieve the residence

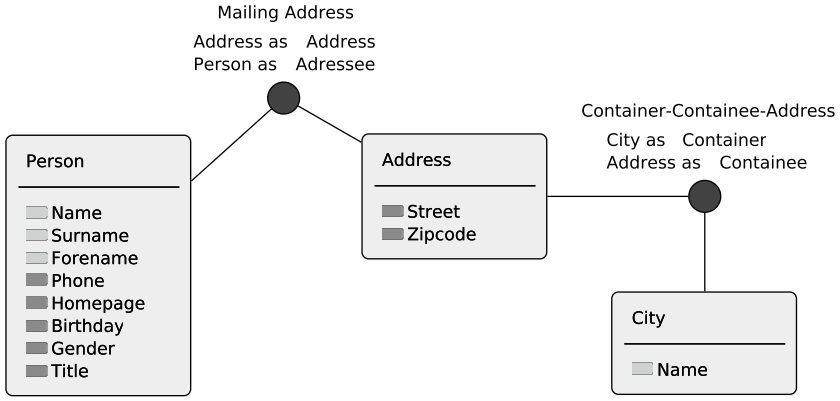



**Fig. 1.** An example model with three Topic types: person, address, and place. In order to model a living-place-facet for a person, a multi-step facet is needed.

of any person, the *indirect association* between the person Topic and its City must be traversed). The authors are aware of the fact, that the chosen example topic map could be modeled in other ways, avoiding indirect associations. Nevertheless, it was extracted from an actually existing project that required a rather fine graining in its model.

This paper describes an easy and standard-based way of defining domain-specific facets for any given Topic. The following section covers the approach of defining domain-specific facets by using the Topic Maps Query Language (TMQL), while Section 3 provides information about the ranking of facets, according to their importance for information needs. Finally, Section 7 concludes this paper.

## 2 Defining Domain-Specific Facets With TMQL

In accordance with the Topic Maps paradigm, any Topic $t$ acts as a symbol to represent one, and only one, subject. Without loss of generality, the set of Topics of any well modeled Topic Map can be divided into two subsets. The set of all Topic types $T$ contains Topics that carry only ontological information about supertype-subtype and instance-of relations, whereas the set of all Topic instances $I$ contains the Topics that hold actual instance data. These two sets may not be distinct, but this is dispensable for our approach. The distinction between ontology Topics and instance Topics is necessary, because facet definitions must be applied to Topic types, whereas the facets themselves can only be applied to Topic instances. A facet $f \in F$ is then defined as the triple

$$f = (i \in I, n, v \in V) \tag{1}$$

where $F$ is the set of all facets, $i$ is an instance, $n$ denotes the name of the facet, and $v$ is the facets value from the set of all facet values $V$.

As aforementioned, facet definitions need to be made at the ontological level of a Topic Map, as defining them for every Topic instance would be inefficient. To define a facet for a Topic type, an expression is needed that allows to effectively precalculate

and assign literal values to instances of any given Topic type. Amongst the family of Topic Map standards, the Topic Maps Query Language (TMQL) is capable of performing this task[1]. Actually, TMQL is much too powerful for this task. The used TMQL implementation TMQL4J[2] provides methods for disabling several parts of the query language, so the set of permissible TMQL expressions is limited to path expressions.

A path expression represents a sequence of navigation steps through the abstract bidirectional graph of a Topic Map. Starting from given values (atoms or items in a Topic Map), navigation steps along defined axes within the context map compute new values [4]. These values can then be filtered according to Boolean conditions and can be used as facet values for the given Topic Map item.

Following the example ontology that is given in Fig. 1, the query to define a living-place-facet for person Topics would be as follows.

```
http://psi.example.com/person
  >> traverse http://psi.example.com/mailing-address
  >> traverse http://psi.example.com/container-containee-address
  >> characteristics tm:name
```

Applying this query to the instances $I_{\text{People}}$ of the Topic type $t_{\text{Person}}$ will yield the name of the city the respective person lives in. This is done by substituting the given subject identifier of $t_{\text{Person}}$ with the subject identifier of every single instance.

## 3 Weighting and Ordering

The use of TMQL path expressions for defining subject-centric facets enables a vast amount of different facets, but not all of them allow for efficient navigation through the data. To identify the highly navigational facets and allow an automatic facet ordering, an automatic mechanism for facet evaluation is needed.

### 3.1 Navigational Value of Facets

A suitable facet allows efficient navigation through the dataset. In this section the "navigational quality" of a facet is defined in terms of three measurable properties of the dataset, following Oren et al. [5]. All values are normalized to $[0 \ldots 1]$ and are combined into a final score through (weighted) multiplication. The metrics provide only an indication of usefulness. Low ranked facets should not be dropped as they could still be intuitive even when totally inefficient.

**Balance** Any tree navigation is most efficient when the tree is well-balanced because each branching decision optimizes the decision power. The balance of any facet $f$ is computed from the distribution $n_i(v_j)$ of the values over the instances as the average

[1] Of course, there is a plenty of languages outside of the Topic Maps world that can fulfill this task. However, the authors ambition was to stay inside the Topic Maps standards family.

[2] Find more information at http://tmql4j.topicmapslab.de/

inverted deviation from the vector mean $\mu$ and is normalized to $[0 \dots 1]$ using the deviation in the worst-case distribution (where $v$ is the total number of different values):

$$\text{balance}(f) = \frac{\sum_{j=1}^{v} |n_i(v_j) - \mu|}{(v-1)\mu + (I-\mu)} \tag{2}$$

**Cardinality** An adequate facet has a limited amount of object values to choose from, typically between two and ten. Too many choices might confuse the user and should necessarily be avoided. The cardinality is computed as the number of different objects $n_0(f)$ for the facet $f$ and normalized using a function based on the Gaussian density depending on the parameters $\eta$ and $\rho$.

$$\text{card}(f) = \begin{cases} 0 & \text{if } n_0(f) \leq 1 \\ \exp\left(-\frac{(n_0(f)-\eta)^2}{2\rho^2}\right) & \text{otherwise} \end{cases} \tag{3}$$

**Frequency** Suitable facets occur frequently inside the collection. The more Topics are covered, the more useful is the respective facet in dividing the information space. The frequency is computed as the number of instances $n_i(f) = |\text{exists}(f)|$ in the dataset for which the facet has been defined, and is normalized as a fraction of the total number of instances $n_i$:

$$\text{freq}(f) = \frac{n_i(f)}{n_i} \tag{4}$$

These three metrics are precomputed and can then be used as objective criteria to improve the quality of the defined facets. This is done by returning the computed values in comparison to recommended standard values.

### 3.2 Use-Paths as Subjective Feedback

While balance, cardinality, and frequency are objective criteria for any kind of facet, they can not take the users subjective *information need* into account. The information need describes the single information a user wants to find inside the data base by walking the shortest possible path[3] from the initial keyword search and through the facet navigation.

However, to find the shortest path one has to take the specific domain into account, how it is modeled in the Topic Map, and to what extend this model differs from the natural understanding that the user has of this domain. It is clear that the user friendliness must be the target of maximization. The behaviour of the user himself is helpful for the task, because he will try to fulfill his information need by walking the path, even if he does not succeed in the first try.

Whenever a user does a keyword search with additional facet navigation, the keywords are stored along with the steps of the facet path that was used. If a significant correlation between a keyword and a distinct facet path occurs, the path can be assumed to be the shortest and most intuitive path to fulfill the information need that is expressed by the correlating keywords. The user interface must then be checked in order to rearrange the displayed facets to meet the users needs.

[3] A path is a sequence of clicks at the user interface.

## 4 Conclusion

Using TMQL path expressions for defining domain-specific facets provides an easy and very flexible way to adopt the paradigm of navigational search for fine-modeled Topic Maps ontologies. The use of three metrics (balance, cardinality and frequency of facets) allows to differentiate between feasible facet definitions and facet definitions that are unusable for navigation through the given data base. Moreover, collected use-paths can be used to adjust the order of displayed facets to afford shorter use-paths for frequently requested information needs and thus an overall better user experience.

## References

1. Tran, T., Mathäß, T., Haase, P.: Usability of keyword-driven schema-agnostic search: A comparative study of keyword search, faceted search, query completion and result completion. In: The Semantic Web: Research and Applications. Volume 6089 of LNCS., Springer (2010) 349–364
2. Taylor, A.G.: Introduction to cataloging and classification. Englewood, Colorado (2000)
3. Ueberall, M., Drobnik, O.: Facet-based exploratory search in topic maps. In Maicher, L., Garshol, L. M., eds.: Subject-centric computing. Fourth International Conference on Topic Maps Research and Applications, TMRA 2008, Leipzig, LIV (2008) 49–62
4. ISO/IEC 18048: Topic Maps Query Language (TMQL). International Organization for Standardization (2008) http://www.isotopicmaps.org/tmql/tmql.html
5. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for rdf data. In: The Semantic Web – ISWC 2006, Berlin/Heidelberg, Springer (2006) 559–572

# Part V

# Investigation Ontology Structure

# External Schema for Topic Map Database

Keita Nabeta[1], Takashi Kojima[2], Yuki Kuribara[1], and Takashi Yamazaki[1]
and Masaomi Kimura[2]

[1] Graduate School of Engineering, Shibaura Institute of Technology.3-7-5,
Koto-ku, Toyosu, Tokyo, 135-8548, JAPAN
[2] Faculty of Engineering, Shibaura Institute of Technology.3-7-5,
Koto-ku, Toyosu, Tokyo, 135-8548, JAPAN
\{m709102,l06048,m109027,m108119,masaomi\}@shibaura-it.ac.jp

**Abstract.** In order to cope with large-scale topic maps that store a lot of information, it is necessary to utilize topic map databases. Although, database management systems should provide users with external schema functions such as views, topic map databases do not have such functions. In this paper, we propose a method of implementing a view function, by focusing on the fact that the substructure of topic maps can be regarded as a topic map. In order to realize the idea, we developed an access control system based on the view function. Through an experiment to measure the execution time, we confirmed that these functions work correctly and have little effect on the execution time.

## 1 Introduction

In order to deal with large-scale topic maps that store a lot of information, it is necessary to utilize a topic map database that provides an efficient data processing method such as update and retrieval. We developed the topic map database system, TOME [1], which is composed of two parts, a query interpreting subsystem and a data manipulation subsystem that manipulates topic map data in *db4o* [2], which is an object oriented database.

If we limit users' access to the data in a database, it is desired that the database has external schema depending on their purpose. Relational databases (RDB) provide us with an external schema, *view*. The view is created by operations such as projections of or selections from real relations, and presented as a table, which has the same structure of a real relation. Users can access records without regard as to whether they are in a real relation or a view.

In this paper, we introduce the implementation of view function to the topic map database. Identifying topics and associations with nodes and edges, we can regard topic maps as graph structures (networks). Taking account of the fact that the substructure of graphs is a graph (sub-graph), we can regard the substructure of topic maps as a topic map.

In order to divide the topic map into substructures, we employed a clustering technique (community detection technique), which is discussed in the field of complex networks. We proposed cluster syntax to extract topic groups from a topic map and implemented the view by limiting the results of queries to the cluster to which a specific topic belongs.

Furthermore, for the case that plural users share the topic map database, it is necessary to permit specific users access to some data even though it must not be disclosed to the public. We propose an access control function by applying the view function discussed above.

## 2 Method

### 2.1 Method to find clusters

As mentioned in the Introduction, we divide the topic map into substructures, which should be obtained as views. In order to find the substructures from topic maps, we apply a network clustering technique proposed by Reichardt and Bornholdt [3]. We extract clusters by optimizing the segmentation of a network, which minimizes the following optimization function:

$$H = \sum_{i \neq j} \left( A_{ij} - \frac{k_i k_j}{2M} \right) \delta(\sigma_i, \sigma_j) \,, \tag{1}$$

where $A_{ij}$ is an adjacent matrix of the network, $k_i$ is a degree of the $i$-th node, and $i$ is the integer number label of the cluster to which the ith node belongs. Regarding this optimization function as Hamiltonian of the spin-glass Potts model usually discussed in condensed matter physics, they also proposed using a simulated annealing method to optimize the segmentation.

We implemented query syntax 'cluster(topicA)', which returns all topics that belong to the same cluster with 'topicA'. The result extracted by the cluster syntax is the list of topics that belong to the same cluster with a topic input as a parameter.

Since finding clusters is an expensive process from the viewpoint of both computational resources and time, we should apply a complex network clustering algorithm in advance of query processing. In processing a query, we make use of the result to identify the cluster to which a topic belongs. In TOME, the management subsystem calls the clustering subsystem a batch process and stores the resultant cluster data as an array list.

### 2.2 Realization of views in topic map databases

In the case of the relational database, the views are realized by appending predicates to limit records whose limitation is given in the definition of the view. Since Topic Map Query Language (TMQL) was defined based on SQL, relational database query language, it is reasonable to realize the view function in the same way as a relational database.

We realized views by AND operation of a given query with the cluster syntax. For instance, when we input the query 'topic-name($TOPIC, $NAME)?', the system appends the predicate 'AND cluster(topicA)?' to it. The given query extracts all topics and their names from the topic map, and the appended predicate limits the topics and their names to the cluster to which 'topicA' belongs.

### 2.3 Implementation of access control system

Next, we implemented the access control system using the view function. In order to limit each user to access specific topics, it is necessary to define the users (and groups) who can login to the database and the topics that each of them can access in advance. Each of the definitions is stored in a 'user list' and an 'authority list'.

The user list includes the set of a user name, a password, a user ID and a group ID. The user name and the password are used for authentication, and the user ID and the group ID are referred to when the system retrieves the condition of topic clusters that the user or the group can access from the authority list.

The authority list includes the set of a user ID or group id, an objective syntax and the predicate that specifies a cluster. The objective syntax has the predicate appended. The user ID or group ID is keys that relate the user/group to the objective syntaxes. If database administrators wish to limit access to part of the topic map to a specific user/group, they can set the user/group ID.

The access control system works in the following steps. First, the system authenticates a user by the user name and password and obtains objective syntaxes and the predicate that specifies the cluster by the user ID and group ID. Next, when the user inputs a query, the system compares it with the objective syntaxes. If the input query matches the objective syntaxes, the system processes the query after the system appends 'AND' operator and the predicate specifying the cluster to it.

## 3 Experiment

We experimented on the performance of the view function and the access control system, and investigated the increase of execution time caused by the addition of access control procedures.

We use two topic maps for this experiment: 'Pokemon topic map' [4] and large-scale random topic map. In the Pokemon topic map, topics express 151 monsters and their type, and associations express their evolution and their relationships to the type. The large-scale random topic map has topics and associations that are randomly generated by WANDORA [5]. Table 1 shows detailed information of each topic map. Table 2 is the user list and Table 3 is the authority list.

**Table 1.** The number of each object

| | Pokemon topic map | Large-scale random topic map |
|---|---|---|
| Topic | 174 | 2,998 |
| Base name | 174 | 2,998 |
| Association | 432 | 9,118 |
| Role | 864 | 18,236 |
| Occurrence | 172 | 0 |

**Table 2.** User list

| User Name | Password | User ID | Group ID |
|---|---|---|---|
| User A | aaaa | 1 | 100 |
| User B | bbbb | 2 | 200 |

**Table 3.** Authority list

| ID | Objective syntaxes | Predicates |
|---|---|---|
| 200 | topic-name | cluster(picachu)? |
| 200 | topic-name | cluster(topic1178)? |

Under these conditions, we evaluated the proposed method. In order to demonstrate the results of the view and the access control function, we executed the query 'topic-name($TOPIC, $NAME)?', which extracts all topics and their names, as a non-limited user (User A) and as a limited user (User B), and compared the results. In order to confirm the correctness of the result for User B, we executed the query 'topic-name($TOPIC, $NAME) AND cluster(picachu)?'.

To verify the affect caused by the addition of access control procedures, we measured the execution time for the query 'topic-name($TOPIC, $NAME)?'. We calculated the average time and its variance of 100 measurements under the following conditions: no access control mechanism, non-limited user (without a predicate) and limited user (with a predicate). We measured the execution time from the time when the system started to the time when it returned the result. In this regard, the user name, the password and the query are automatically input in the measurement program.

## 4 Results and Discussion

### 4.1 Demonstration of the results of the view and the access control function

Figure 1 and Fig. 2 show the results obtained by the non-limited user (User A) and the limited user (User B). In the former, all topics and their names were extracted from topic maps. In the latter, since User B is limited to access only the cluster to which picachu belongs, topics and their names in the cluster were extracted, which is the expected result for the access control system.

Next, we illustrate the result that was returned from a query 'topic-name($TOPIC, $NAME) AND cluster(picachu)?' (Fig. 3). Since the result for User B is the same as this result, we can see the correctness of it.

### 4.2 Verifying affect of execution time caused by implementation of our method

Table 4 shows the number of rows, average execution time and their variance under the conditions: no access control mechanism, non-limited user (without the predicate) and limited user (with the predicate).

```
Input your user name and password
User name: User A
Password: aaaa
You succeeded to access database
Select Topic Maps: queryTM(Poke.db4o.pokmeonTM)
Query: topic-name($TOPIC, $NAME)?
Rows: 174
  $TOPIC = bulbasaur    $NAME = bulbasaur
  $TOPIC = ivysaur      $NAME = ivysaur
  $TOPIC = venusaur     $NAME = venusaur
  $TOPIC = charmander   $NAME = charmander
  $TOPIC = charameleon  $NAME = charameleon
        .                     .
        .                     .
        .                     .
  $TOPIC = reside       $NAME = reside
  $TOPIC = group        $NAME = group
  $TOPIC = monster      $NAME = monster
  $TOPIC = pokemon      $NAME = pokemon
  $TOPIC = instance-of  $NAME = instance-of
```

**Fig. 1.** Result returned by query without access control (User A)

```
Input your user name and password
User name: User A
Password: aaaa
You succeeded to access database
Select Topic Maps: queryTM(Poke.db4o.pokmeonTM)
Query: topic-name($TOPIC, $NAME)?
Rows: 174
  $TOPIC = bulbasaur    $NAME = bulbasaur
  $TOPIC = ivysaur      $NAME = ivysaur
  $TOPIC = venusaur     $NAME = venusaur
  $TOPIC = charmander   $NAME = charmander
  $TOPIC = charameleon  $NAME = charameleon
        .                     .
        .                     .
        .                     .
  $TOPIC = reside       $NAME = reside
  $TOPIC = group        $NAME = group
  $TOPIC = monster      $NAME = monster
  $TOPIC = pokemon      $NAME = pokemon
  $TOPIC = instance-of  $NAME = instance-of
```

**Fig. 2.** Result returned by query with access control (User B)

```
Select Topic Maps: queryTM(Poke.db4o.pokmeonTM)
Query: topic-name($TOPIC, $NAME) AND
cluster(picachu)?
Rows: 10
  $TOPIC = raichu       $NAME = raichu
  $TOPIC = picachu      $NAME = picachu
  $TOPIC = magnemite    $NAME = magnemite
  $TOPIC = magneton     $NAME = magneton
  $TOPIC = voltorb      $NAME = voltorb
  $TOPIC = electrode    $NAME = electrode
  $TOPIC = jolteon      $NAME = jolteon
  $TOPIC = electric     $NAME = electric
  $TOPIC = electabuzz   $NAME = electabuzz
  $TOPIC = zapdos       $NAME = zapdos
```

**Fig. 3.** Result of view created by cluster syntax

**Table 4.** Result of execution time

| Topic map | Condition | Rows | Average time (mill second) | Variance |
|---|---|---|---|---|
| Pokemon Topic map | No access control mechanism | 174 | 1,488.61 | 6,231.78 |
| | Non-limited user | 174 | 1,696.60 | 1,490.64 |
| | Limited user | 10 | 1,717.19 | 1,526.25 |
| Large-scale random topic map | No access control mechanism | 2,998 | 3,293.59 | 4,457.58 |
| | Non-limited user | 2,998 | 3,579.76 | 11,338.78 |
| | Limited user | 264 | 3,580.00 | 3,040.02 |

Regardless of the size of topic maps, the difference of execution time without the access control mechanism, time to process the query for a non-limited user and for a limited user comes from the existence of the user authentication and the view functionality. Since there is no difference between them, at least in the leading order of the execution time, the user authentication does not affect the execution time for a topic map that has about 3,000 topics.

## 5 Conclusion

In this study, we proposed a view function and an access control function for a topic map database.

In order to specify the group of topics to which users are limited access, we developed a cluster syntax that returns topics that belong to the same cluster by applying a network clustering algorithm to the topic map. By appending the 'AND' operator and the cluster syntax to the given query, we realized the external schema (view) of topic maps, taking account that the substructure of a topic map can also be regarded as a topic map. Based on this view functionality, we realized the access control mechanism by defining and matching the user list and the authority list.

The results of the experiment confirmed that these functions work correctly and that there is only a small increase on execution time caused by the addition of the access control mechanism for topic maps that have about 3,000 topics.

In this paper, since we implemented an access control function for only retrieval operation, we will apply it to insertion and deletion operations in future work. Furthermore, we will discuss the way of defining topic groups other than network clusters for limiting user access.

## References

1. Yuki Kurabara, Takeshi Hosoya, Masaomi Kimura: TOME: Topic Maps Database Extended. The 4th South East Asian Technical University Consortium (SEATUC) Symposium. pp.245—248 (2010)
2. Versant Corporation: db4objects, http://www.db4o.com/

3. Joerg Reichardt, Stefan Bornhold : Statistical mechanics of community detection, Physical Review E, vol. 74, 016110, pp.1–14 (2006)
4. Pokemon Topic Map, http://www.ontopia.net/omnigator/models/topicmap_complete.jsp?tm=pokemon.ltm
5. WANDORA, http://www.wandora.org/
6. Motomu Naito: An Introduction to Topic Maps. Tokyo Denki University Press (2006)
7. Ontopia: tolog Language tutorial, http://www.ontopia.net/
8. ISO/IEC JTC1/SC34, Topic Map – Data Model, http://www.isotopicmaps.org/sam/sam-model/

# Evaluation of Instances Asset in a Topic Maps-Based Ontology

Petra Haluzová

Department of Informatics and Telecommunications
Faculty of Transportation Sciences
Czech Technical University in Prague
11000 Prague, Czech Republic
haluzpet@fd.cvut.cz

**Abstract.** This short paper presents the methodology of information asset evaluation of individual instances in ontology based on the standard of Topic Maps. The evaluation is derived from weight assignment to instances and associations in ontology and total weight calculation for an individual instance and its surroundings. Search results can be ranked according to this total weight expressing the information asset of individual instance.

## 1 Introduction

In connection with knowledge evaluation, specifically their gains evaluation and currently discussed ontologies, the question arises as to what the ontology actually yields. This short paper aims to find a way to quantify the information asset of concepts (instances) in the ontology. The information asset expresses the richness of instance description and thus points to its potential usefulness for the user. The paper proposes a methodology by which the total weight of the information asset is calculated for each instance, also with respect to its surrounding instances. Instances found during a search in the ontology may be ranked according to this score and therefore the instances described richly and in detail can be placed at top of the list of search results. Considerations and calculations are performed in relation to an ontology created in Topic Maps (TM), but the main idea is applicable in general terms to other standards as well.

The paper is organized as follows: Section 2 presents the work related to the theme of ontology or their parts evaluation. Section 3 describes weights assignment to instances and associations and proposes the entire procedure of evaluation. The last section contains conclusions and several notes of further work possible.

## 2 Related Research

Various terms are used in this section, depending on the authors. Terms such as relation, relationship, and association, as well as concept, class, entity and instance, can be considered as equivalent, depending on the context.

Zouaq and Nkambou [1] mentioned that there is no single right way of evaluating ontologies. It is more practical to focus on the evaluation of different levels of the

ontology separately rather than trying to evaluate the ontology as a whole. The levels have been defined variously depending on authors, e.g.: lexical or data layer; hierarchy; semantic relations; context, application level; syntax; structure, design [2].

There follows an approach summary of several authors related to RDF and OWL ontologies evaluation based on their structure with examples of various metrics and weights implementation. The authors in [3] define a set of cohesion metrics. The proposed metrics are Number of Root Classes (without any semantic super classes), Number of Leaf Classes (having no semantic subclasses) and Average Depth of Inheritance Tree of all Leaf Nodes.

Huang and Diao [4] put forward six properties for ontology structure characterization. The approach is based on the internal ontology structure, statistics and graph theory. The proposed properties are: Concept Quantity (corresponds to the size of ontology), Property's Expectation (provides an overview of the abundance of relations among concepts), Property's Standard Deviation (reflects the situation of the property's distribution in ontology), Key Concept Quantity (expresses the number of concepts which have more adjacent concepts than the other) etc.

The authors in [5] introduce Weight Mapping – a technique for calculating the numerical weight value for each relation, based on the analysis of the knowledge base structure. It helps to obtain better search results, because all concepts instances that are related to a given keyword are also taken into account. Three different measures were proposed in order to calculate the weights (Cluster Measure, Specificity Measure and Combined Measure).

Semantic Associations (relationships between entities) are discussed in [6]. The semantic association is represented as a path in an RDF graph. This path consists of classes (instances) and their inter-relationships. The authors assigned two types of weights to each path: Universal and User-Defined. The Universal Weight (called Subsumption Weight) consists of weights of each instance occurring in the path. The weight of an instance comes from its position in a hierarchy. The lower instances in the hierarchy can be considered to be more specialized and therefore their weight has a higher value. The User-Defined weights are Path Length Weight (a longer path may infer possibly hidden and indirect relationships), Context Weight (a user can assign weights to particular regions of ontology, i.e., to sub-graphs in the RDF graph) and Trust Weight (some information sources may be trusted more than others). So, we obtain the total weight of each path which corresponds to the quantitative evaluation of the semantic association between two concepts.

The approach presented in this paper does not consider concepts alone, but also their surroundings depending on relation type, similarly as in [5]. As in [6], the user's insight is taken into consideration, if desired (because the entire process can be simplified by automated weight assignment).

## 3 Methodology of Asset Evaluation of Instances

### 3.1 Ontology Matrix Structure

The ontology in TM consists of three main elements: Topics, Associations and Occurrences [7]. Basically, it is possible to consider the associations and occurrences as

attributes, which can gain particular values, and they are added to topics. The attribute-association value is another topic in TM (but we do not consider associations as attributes in this paper). The attribute-occurrence value can be, for example, www URL, article, reference to specific part of text, concise or detailed description directly in topic, picture etc. Topics can also comprise of other attributes describing their properties, characteristics or parameters.

Each of these three main elements can have its own type and these types become other topics in the same TM. It results in the fact that everything in the TM becomes a topic. Nevertheless, it is important to distinguish between topic type and instance. The instance is a final element which cannot contain other instances. In contrast, the topic type can include an instance. This short paper discusses the asset evaluation of instances. Therefore, we consider only the selected (listed) relationships (divided into categories – see section 3.2.2) and not the generally (automatically) established ones (e.g., '*has cardinality*', '*use edit mode*'). What we get is a list of topics among which the selected relationships exist. The proposed methodology applies only to ontologies with a large number of instances where the topics thus found (or at least the majority of them) could be regarded as instances.

Numerical or string IDs are automatically assigned to all elements in XTM. There may appear different topics under the same name in TM – but different IDs are adhered to them.

Figure 1 shows an example of ID#98 instance connection with surrounding instances. As we can see in this scheme, it arises that we need all instances and their attributes (Fig. 2) and all interconnected instances including association types (Fig. 3). This relevant information for calculation of the information asset is extracted from XTM to matrix notation. Specifically, these are topic IDs, occurrence types present in particular topics

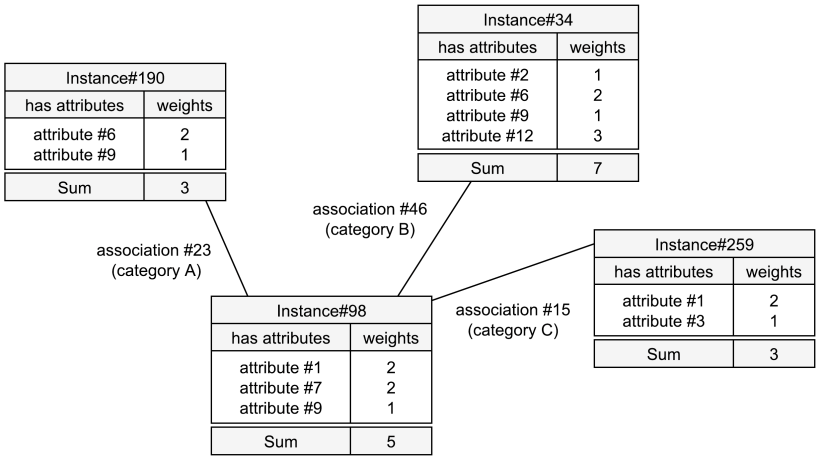



**Fig. 1.** Scheme of instances interconnection and attributes weights assignment

| instance id | attribute id 1 | 2 | 3 | 6 | 7 | 9 | 12 | Sum |
|---|---|---|---|---|---|---|---|---|
| 34 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 7 |
| 98 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 5 |
| 190 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 3 |
| 259 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 3 |

**Fig. 2.** Matrix of instances and their attributes

| instance id | instance id 34 | 98 | 190 | 259 |
|---|---|---|---|---|
| 34 | | 46/B | 0 | 0 |
| 98 | 46/B | | 23/A | 15/C |
| 190 | 0 | 23/A | | 0 |
| 259 | 0 | 15/C | 0 | |

**Fig. 3.** Matrix of instances and their associations

(called attributes here) and association types between topics. Then we can assign partial weights and process the calculation for each individual instance. Hence this matrix (or list) notation of ontology facilitates the algorithmic processing, i.e., total weight calculating and recalculating according to partial weight changes. Jin and Liu [8], for example, also viewed and used ontology as a matrix.

### 3.2 Instances and Associations Weights

**Partial Weight of Instance** Partial weight of each instance $= \{1, 2, \dots\}$; $c \in N$ emerges from the description in ontology, i.e., how many and which attributes the instance contains (Fig. 2). The user individually assesses each attribute according to its information asset. For example attribute-occurrence '*picture'* has greater importance than attribute-occurrence *'concise description'*. In contrast, attribute-occurrence '*external web link'* has a higher weight value than both of the aforementioned attributes, because we assume that the webpage offers more information (than the picture or concise description), and therefore it has a greater information asset. There are many attribute types depending on the size and elaboration of ontology, so we can simplify weight assignment by attributing 1 to them. It means that we perceive all attributes at the same importance level, and the partial weight of instance grows evenly depending only on the attribute amount, but not depending on its type.

**Association Weight** The weight of each association $a \in \langle 0, 1\rangle$; $a \in R$ is determined by the category to which the association belongs. It appears appropriate to put associations into three categories (A, B, C in Fig. 3):

- Hierarchical – express the hierarchical order, i.e., the relationship between more general and less general instances.
- Defining – refer to definitions, origin, explanations. They are the strongest of all these categories.
- Contextual – express contexts, examples, notes. They are the weakest of all these categories.

The categorization of associations is difficult to generalize because it depends on the specific ontology and on the user's point of view. For example, in the sample ontology

Opera (from Ontopia tools), which contains composers, their birthplaces, works etc., we consider the association *'was learner-teacher'* as hierarchical, *'based on'*, *'composed by'* as defining and finally *'has voice type'* as contextual. We can simplify category weight assignment by attributing the same value to them, i.e., we are interested in the existence of an association between two instances, not in the association type.

It would be possible to assign the weight of each association type as in case of instance attributes and then normalize these weights to interval $\langle 0, 1\rangle$. However, associations are not primarily as important for information asset evaluation as attributes are, therefore it is sufficient to divide them into groups and assign weights according to these groups.

**Total Instance Weight** From the above mentioned weights we calculate the evaluation $w_i$ of each individual instance $i$ in ontology as:

$$w_i = c_i + k \cdot \sum_i a_{ij} \cdot c_j \tag{1}$$

where $c_i$ is a partial weight of instance $i$, $a_{ij}$ are weights of associations between instance $i$ and surrounding instances $j$, $c_j$ are partial weights of surrounding instances $j$. The coefficient $k \in \langle 0, 1\rangle$, $k \in R$ expresses the weight we attribute to all surrounding instances (independently of partial weights of these instances). Unless we want partial weights of surrounding instances $c_j$ to influence the total instance weight $w_i$ to a great extent, we choose e.g. $k = 0.5$. In the event that the partial weights of surrounding instances are of the same importance as a partial weight of central instance (instance $i$), we ascribe $k = 1$.

Let us consider the example in Fig. 1. Because of the simplification we use the same weight values for all association categories, i.e., $a_{ij} = 1$ and $k = 0.5$ (we perceive partial weights of surrounding instances as having a half level of importance in comparison with the central instance). The total instance weight # 98 according to (1) is:

$$\begin{aligned} w_{98} &= c_{98} + k \cdot \left[(a_{98.34} \cdot c_{34}) + (a_{98.190} \cdot c_{190}) + (a_{98.259} \cdot c_{259})\right] \\ &= 5 + 0.5 \cdot (7 + 3 + 3) = 11.5 \end{aligned} \tag{2}$$

After calculating the total instance weight for each individual instance in ontology we normalize all results using the maximal calculated value to interval $\langle 0, 1\rangle$.

### 3.3 Methodology of Asset Evaluation of Instances

We summarize the mentioned procedure in steps as follows:

- Assign the weights to all attributes which can occur within an instance (simplification: consider weight of all attributes 1).
- Divide associations into three categories introduced in Section 3.2.2. and assign the weights to these categories (simplification: consider weight of all associations 1).
- Calculate the total instance weight for each individual instance in ontology according to (1).
- Normalize all calculated total instance weights in accordance with the maximal value.

## 4 Conclusion and Further Work

This short paper introduced the simple procedure of obtaining the information asset quantification for individual instances in ontology. We assigned the weights to individual concepts (instances) and associations. Based on these weights and on the relation of topics via associations we calculated the total evaluation value of each individual instance and its surroundings. If several instances correspond with the search criterion, we can rank the search results according to this calculated value. So far the analyses have been carried out in a testing ontology created from several transport standards. In addition, the sample ontology Opera from the Ontopia tools was used.

Further work will be focused on the proving of the method functionality through experiments and on the automation of the whole calculating process. The extension of relevant surroundings of each instance will be taken into account. The partial weights of instances in more distant surroundings could be decreased as appropriate by means of further coefficient.

## References

1. Zouaq, A., Nkambou, R.: Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project. IEEE Transactions on knowledge and data engineering, Vol. 21, Issue 11 (2009)
2. Brank, J., Grobelnik, M., Mladenic, D.: A survey of ontology evaluation techniques. In: Proceedings of IS'05, Ljubljana (2005)
3. Yao, H., Orme, A. M., Etzkorn, L.: Cohesion Metrics for Ontology Design and Application. Journal of Computer Science, Vol. 1, Issue (2005)
4. Huang, N., Diao, S.: Structure-Based Ontology Evaluation. In: Proceedings of the IEEE Int. Conf. on e-Business Engineering, Shanghai (2006)
5. Rocha, C., Schwabe, D., Aragao, M.: A Hybrid Approach for Searching in the Semantic Web. In: Proceedings of WWW Conference'04, New York (2004)
6. Aleman-Meza, B., Halaschek, Ch., Arpinar, I., Sheth A.: Context-Aware Semantic Association Ranking. In: Proceedings of SWD'03, Berlin (2003)
7. Pepper, S.: The TAO of Topic Maps, Ontopia (2002)
8. Jin, L., Liu, L.: An Ontology Definition Metamodel based Ripple-Effect Analysis Method for Ontology Evolution. In: Proceedings of CSCWD '06, Nanjing (2006)

# Topic Maps Graph Visualization & Suggested GTM

Rani Pinchuk and Jelle Pelfrene

Space Applications Services
Leuvensesteenweg 325, 1932 Zaventem
\{rani.pinchuk,jelle.pelfrene\}@spaceapplications.com
http://www.spaceapplications.com

**Abstract.** The goal of this work is to generate graphical representations that visually present knowledge kept in a topic map, in a way that is best grasped by users. The paper first describes an idea of providing a contextually relevant excerpt of the topic map as a graph. Then the graphical notation for representing these excerpts is discussed.

## 1 Introduction

Topic Maps allows externalizing knowledge by documenting the relationships between concepts. Documenting these relationships – the associations between topics and the occurrences of topics – can help the user to quickly understand the ideas stored in the topic map.

However, in order to reach such a goal, the topic map must be visualized in a way that is natural to the user and that keeps the original meaning of these relationships.

Presenting the full topic map to the user is not useful as the user will be overwhelmed with data – a topic map is actually a cloud of associated topics. Instead, a contextually relevant excerpt of the topic map can be provided to the user.

The context is defined by the user choice – by selecting certain Topic Maps items. The excerpt provided includes these chosen items together with other Topic Maps items which are semantically close (in the topic map context) to the selected ones. We call such a topic map excerpt a Reduced Graph.

Section 2 below describes what such a Reduced Graph should contain. Section 3 presents the visualization of the Reduced Graph.

## 2 The Reduced Graph

### 2.1 Topic Map as a Graph

A specific topic map can have different graph representations. The main goal in designing our custom graph-based representation is to preserve the semantic distances as provided by the original topic map. That is, although the type of an association is defined as a topic by itself, and although a role type in an association is defined as a topic as well, our graph does not directly connect the players of different associations to the nodes representing those typing topics as shown in the left side of Fig. 1.

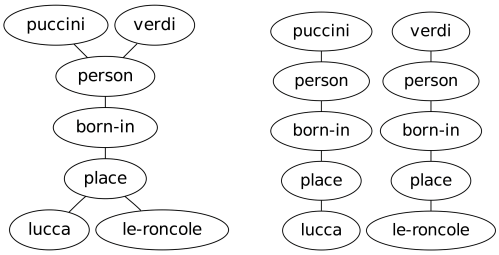



**Fig. 1.** Different representations of associations

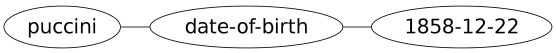



**Fig. 2.** Occurrence chain example

Instead, for each association type, and role type, a distinct node is created, as if the topics defining these types were duplicated – as shown in the right side of Fig. 1.

Based on this approach we define the graph as follows:

Each topic is represented as a node in the graph. One topic may be represented in the graph more than once.

An occurrence-value is represented as a node in the graph. An occurrence of a certain topic is presented in the graph as a *chain* of three nodes connected by edges (as shown in Fig. 2):

1. The node representing the topic containing the occurrence.
2. The node representing the topic which types the occurrence.
3. The node containing the occurrence-value.

Type-instance relationship as found in the topic map is defined as a relationship where the parent is the type, and the child is the instance. Similarly, supertype-subtype relationship is defined as a relationship where the parent is the supertype and the child is the subtype. In our graph, each topic is recursively chained to its parents until no further parents are found as demonstrated in Fig. 3.

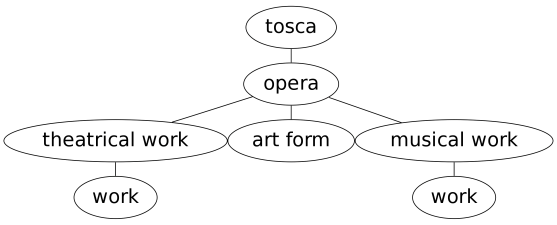



**Fig. 3.** Types chain example

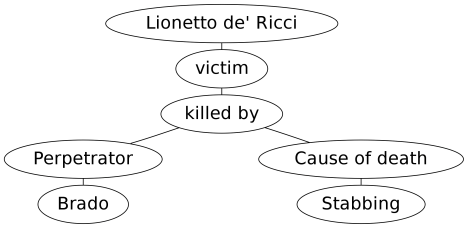



**Fig. 4.** Ternary association chain example

Other associations than type-instance or supertype-subtype are built as a *chain* that looks like a star. The node representing the association type is placed in the middle of the star. All roles are connected to the central node and to the associated players.

In Fig. 4 one association of type *killed by* is presented. *Lionetto de' Ricci* plays the role *victim* in the association, *Brado* plays the role *Perpetrator* and *Stabbing* plays the role *Cause of death*.

Finally, if there are two nodes in the graph which refer to the same topic and those nodes are not used as association types, association roles or occurrence type, they should be merged. All the edges connected to each one of them will be connected to the merged node.

That is, after the nodes in the graph are merged, the only topics that can have multiple nodes representing them in the graph are topics that are used as association types, association roles or occurrence types.

### 2.2 The Reduced Graph

We name the Topic Maps items selected by the user *anchors*. The excerpt we look for should be the smallest sub graph which contains all the anchors.

Mathematically, this problem is very similar to the Steiner tree problem [1] which, in its simplest form can be explained as follows: given a graph in which a subset of vertices are identified as terminals, find a minimal connected sub-graph that includes all the terminals.

The Steiner tree problem is NP complete. This suggests that finding the Reduced Graph is a challenge too hard to be solved – especially in a scalable manner. However, we stress that the anchors are selected by the user. This means that when the user has a certain idea he wants to explore, the anchors will be related in his mind and closely related semantically in the topic map.

This realization, that the anchors must be close to each other, allows us to terminate the search process when it has become too deep, therefore confining the complexity issue to a scalable level.

Unlike the Steiner Tree, the Reduced Graph should include *all* the shortest links between two anchors. This way, when the user expresses interest in “character”, “bass” and “Tosca”, a suitable topic map excerpt should contain the elements shown in Fig. 5. It can be seen that between the anchor *tosca* and the anchor *bass*, there are more than one path possible, all equally interesting.

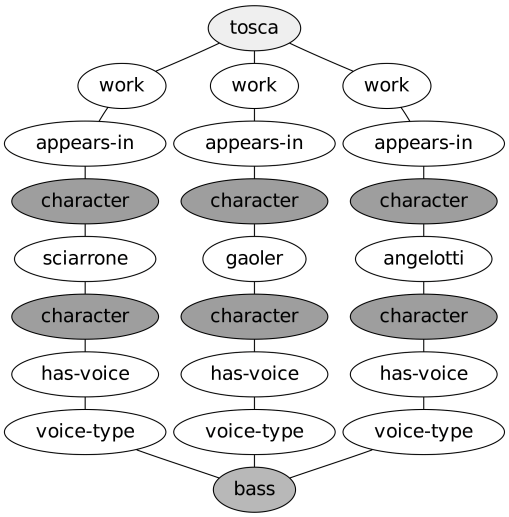



**Fig. 5.** Reduced Graph for Tosca, character and bass

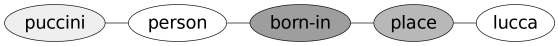



**Fig. 6.** Reduced graph for puccini, born-In, place

Moreover, because we aim at showing the user a useful context in the excerpt, it is important to include the full associations or occurrences chains. This is well demonstrated in Fig. 6. The shown association chain connecting *puccini* and *lucca* contains 5 nodes in total. However, only 4 nodes are contained in the Steiner tree for the anchors *puccini*, *born-in* and *place*. The relevant topic *lucca* falls just outside of the Steiner tree.

Therefore, a Reduced Graph should contain full *chains* (as defined in section 2) and not parts of chains.

## 3 Topic Map Graph Visualization

So far we have shown how Topic Maps can be modeled as a graph, and how a relevant sub set of it can be defined according to the context provided by the user. In order to present this found Topic Maps excerpt, we can in theory use the entire plethora of generic graph visualization algorithms. However, for a graph derived from a topic map, we know that, semantically, not every node or edge is equally important. This indicates that a more specific visualization is worth pursuing.

One suggestion for such a Topic Map graphical notation is GTM$^{\text{alpha}}$ [2]. In GTM$^{\text{alpha}}$, the “everything is a topic” principle is stressed in order to visually demonstrate that the types of roles, associations, occurrences and names as well as scopes are all topics. This leads to a representation that includes graphical links between for example, roles of different associations to the same topic. As a result, the reader of the graphical

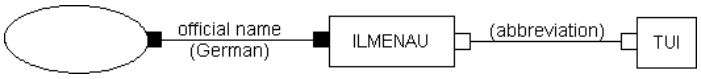



**Fig. 7.** Name and variant

representation gets a picture where the most important relationships in the topic map are no longer visually distinct.We suggest avoiding marking an association type, role type, occurrence type, name type or a scope by linking those to the actual topic with a line. Instead we suggest having a text above or below the line, and the meaning of that text is that it *refers* to the typing topic. If it is a scope, we suggest having that text with a bracket around it.The figures below, which are based on the figures from [2], demonstrate this idea. In Fig. 7, we can see a topic name "ILMENAU" which is of type "official name", in the "German" scope, and which has a variant in the scope "abbreviation".

In Fig. 8, the occurrence "98544" is typed by "zip code" and scoped by "Germany".

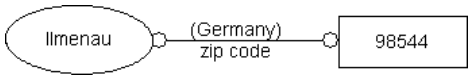



**Fig. 8.** Occurrence

In Fig. 9, the two topics "Ilmenau" and "Thuringia" are associated in association of type "is-part-of", and this association is scoped by "Germany".

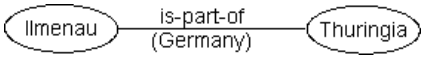



**Fig. 9.** Association

In Fig. 10, the roles in the last association are also mentioned.

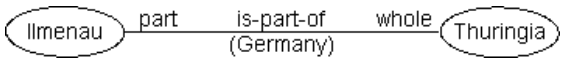



**Fig. 10.** Association roles

The comparison below demonstrates the difference between the two suggested approaches.

Because the visual representation in Fig. 12 reflects better the semantic distances, the meaning of the relationships in this representation is clearer.

Especially for the novice users who are barely familiar with Topic Maps, the "everything is a topic" principal is confusing in the context of such presentations. We believe

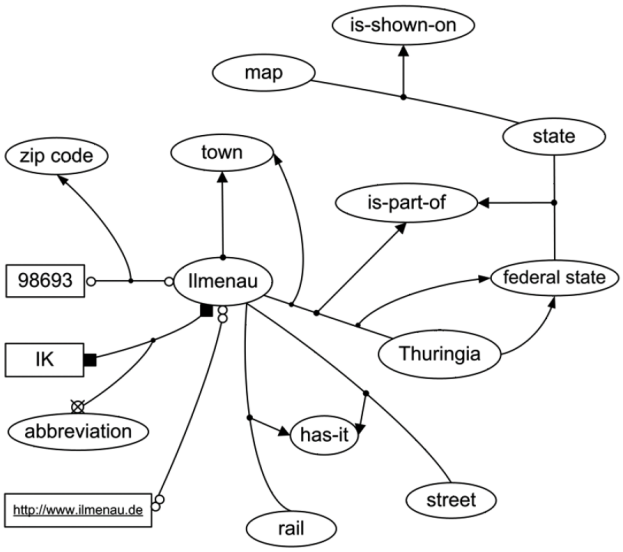



**Fig. 11.** Domain view of the topic map draft in GTM[alpha] (taken from [2])

that a main usage of GTM will be to communicate ideas to domain experts who are not familiar with Topic Maps. For those users, a simple line between two ovals is translated almost naturally to an association between two topics.

Another reason to support this suggestion is that it prevents the situation where scopes and topics which type associations, roles, occurrences or names become graphical "hot spots" (that is, many lines go out of those topics – such as "is-part-of" or "has-it" in Fig. 11). In our experience, the users are more interested in players in associations that are "hot spots" (such as "Ilmenau" in Fig. 12).

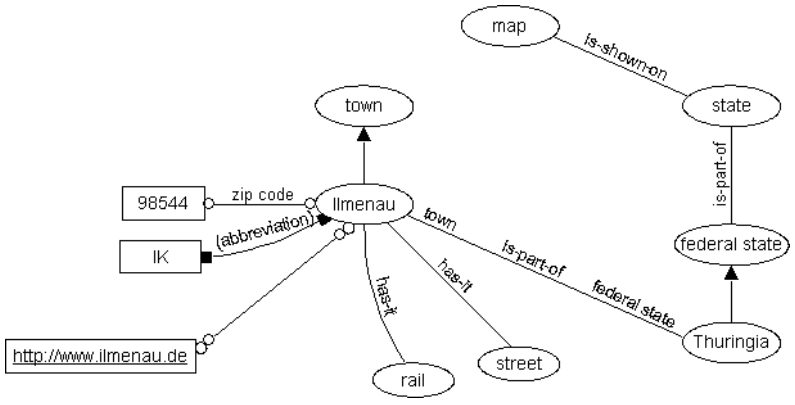



**Fig. 12.** The same domain view in GTMalpha with the suggested change

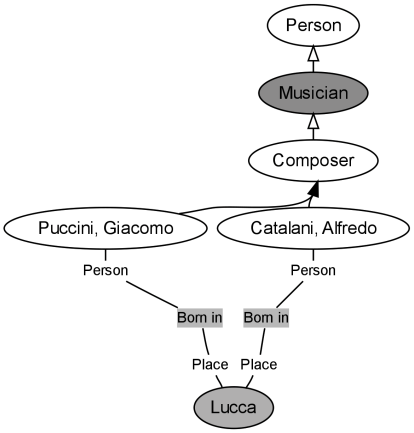



**Fig. 13.** Topic Maps excerpt for Musician, Born in and Lucca

Finally, another variation on GTMalpha is the graphical distinction between supertype-subtype relationship and type-instance relationship. It is suggested that supertype-subtype relationship is notated with an unfilled arrowhead as opposed to the filled arrowhead for type-instance relationship. Figure 13 above presents a generated topic map excerpt illustrating this distinction.

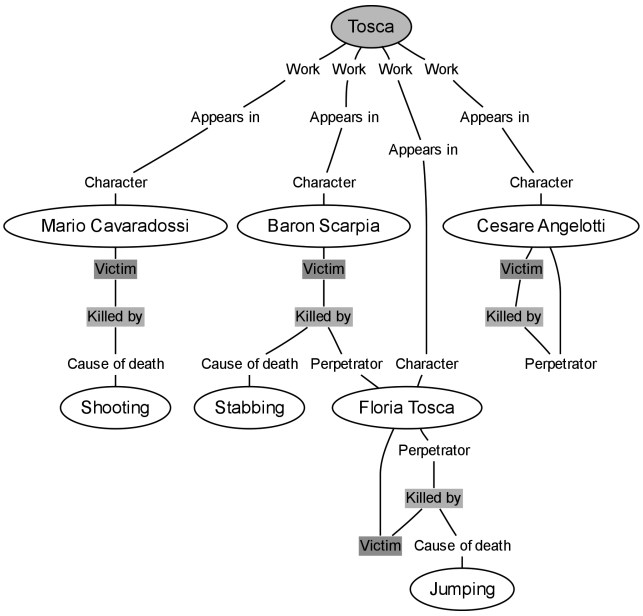



**Fig. 14.** Topic Maps excerpt for Tosca, victim and Killed by

Figure 14 is a generated excerpt that illustrates the readability of this kind of presentation. Even users who are not at all familiar with topic maps can understand, for example, that Floria Tosca stabbed Baron Scarpia and jumped to her death.

## 4 Related Work

The comparison between this proposal and the GTMalpha proposal is worked out in Section 3. Other proposals were documented in a survey[3] as part of the GTM process.

The GTM standardization process has defined two levels of graphical notation: the ontology level and the instance level. Like GTMalpha, this proposal falls under the instance level.

A more elaborate classification of Topic Maps visualization techniques is offered by Benedicte le Grand[4]. She also distinguishes between two sets of requirements for visualizations: either for a representational or a navigational purpose. The graph visualization we defined tries to fit the representation role in a user-friendly manner. Our definition of a Reduced Graph for a specific user-defined context crosses over to the navigation use-case, by focusing the contents represented on a specific user-selected part of the global topic map.

## Acknowledgments

This work has received funding from the European Commission through the Seventh Framework Programme (FP7/2007-2013) under the Space Theme, under grant agreement n°218815 within the ULISSE Project (USOCs KnowLedge Integration and Dissemination for Space Science Experimentation).

## References

1. Magnús M. Halldórsson et al.: Approximating Steiner trees in graphs with restricted weights, in: Networks 31(4):283–292, 1998
2. Hendrik Thomas et al.: GTMalpha – Towards a Graphical Notation for Topic Maps, in: 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
3. Garshol, L.M., Lee, J.: Survey of graphical Topic Maps notations, on, http://www.isotopicmaps.org/gtm/survey.html, 06-Jul-2007
4. Le Grand, B., Soto M.: Visualisation of the Semantic Web: Topic Maps Visualisation, in: IEEE IV 2002, London, July, 2002

# Part VI

# Semantic Integration

# A new approach to semantic integration

Lars Marius Garshol

Bouvet ASA, Oslo, Norway
larsga@bouvet.no, http://www.bouvet.no

**Abstract.** This paper presents an approach to using semantic technologies to achieve better and more flexible integration of IT systems. The author believes that the described approach is applicable to a great many organizations, and that it can lead to far more dynamic IT architectures than what is common today.

## 1 Introduction

Many organizations lack strong central control over their IT systems, and do not have clear principles and goals for the growth of internal IT systems. Systems and projects therefore grow haphazardly and in different directions, and integration between systems is generally solved on an "ad hoc" basis for each individual case. The result is inevitably a rather anarchistic systems structure.

A contributing factor to these problems is the mergers, acquisitions, and reorganizations that have been so common in the private and public sectors over the last decades. A side-effect of these changes is that IT systems move along with the employees that serve them, and so architectures that made sense when originally designed may no longer do so when the organization changes.

A central problem in these architectures tend to be a failure to encapsulate data and business logic, which leads to multiple systems being integrated directly against the same raw data, causing the same business logic to be reimplemented in multiple systems. This is a risky strategy, as there is every chance that the business logic in the different systems fails to be exactly identical, and there is a greater chance of errors in each reimplementation. It is also expensive, since several different components performing the same function must be maintained in parallel.

Another common problem is a failure to centralize master data, which means that data about a single entity is duplicated in different systems, and therefore has to be maintained in several places. This greatly complicates not just data maintenance, but also ordinary use of the IT systems, and can hamper an organization's day-to-day work significantly [Economist10].

Service-oriented architecture (SOA) and Enterprise Service Buses (ESBs) were the previous decade's proposed solutions to these problems, often combined with attempts to enforce strict central control over architecture. The result was often an over-emphasis on technology as the solution, and central oversight putting a damper on the implementation of practical solutions throughout the organization. ESBs in particular, but also to some degree SOA, are therefore today coming under heavy criticism from many quarters [McKendrick08].

So what should be done? Parts of the solution are non-technological. Organizations need to find a balance between central dictatorship and total anarchy, and actively manage their information and systems architectures. Changing the role of central IT architects from a restrictive policing role to a more supportive advisory role can help substantially. Similarly, establishing sensible best practices for creating and maintaining web services helps [Hinchcliffe09].

Of course, the solution must also be partly technological, and the basic ideas of SOA remain as relevant as ever, but it's clear that the JBOWS approach (Just a Bunch of Web Services) is not sufficiently coordinated. What is really needed is a more flexible and dynamic approach, and this is where semantic technologies can help.

In this paper we describe a possible approach to solving these problems, based on semantic technologies.

## 2 Step 1: Mapping the landscape

The first step on the road to a better information architecture is to build a map of the internal IT landscape. That is, to map the IT systems and web services that exist in the organization. We propose doing this by describing the systems, the web services, and the data they manage using Topic Maps. Several authors have already described projects that have built such maps [Brady04] [Gulbrandsen03], but we propose taking the approach further than what has already been described.

The topic map needs to describe the following:

- The internal IT systems, with names and descriptions.
- The internal data models of these systems, with entities and the fields on each entity.
- The different web services, with names, descriptions, end point URLs, and the IT systems they are based on.
- Which entities and fields are available from each web service.

Below we give a very simplified example, using CTM syntax:

```
pr - "Payroll system";
  isa it-system.
ac - "Accounting system";
  isa it-system.

pr-employee - "PR_EMPLOYEE";
  isa entity;
  belongs-to-system(pr).
pr-employee-name - "NAME";
  isa field;
  belongs-to-entity(pr-employee).

pr-employee-data - "GetEmployeeData";
  isa web-service;
  based-on(pr);
  used-by(ac);
  can-return(pr-employee);
  can-return(pr-employee-name).
```

Actually creating a topic map of the schema of large internal systems may be a time-consuming task, but it can be simplified by scripting, since the schemas of for example relational databases can easily be queried in SQL.

Creating such a topic map will provide a useful overview of the internal IT systems, but perhaps less of an overview than one might think at first glance. The problem is that there is no connection between the entities in the different systems. So while the topic map may show that three different IT systems have the entity `APPLICATION` there is nothing to show what the relationship between these three entities really is. That is, it will not be clear whether these entities represent the same thing, closely related things, or things which are actually not related at all. And other IT systems may have the same or closely related entities under different names without this being made clear.

Therefore an internal reference data model must be developed. This is a simplified high-level data model describing central concepts in the organization's information, and the main properties and relations on these concepts. The reference data model must be independent of any specific IT system in the organization, and should reflect the concepts used by employees as closely as possible. The model will have to be built, owned, and maintained by the organization itself as far as possible, since the model will need to be a living model, adapting and changing as the organization itself changes.

The reference model is also to be expressed in the topic map, as shown in the CTM example below:

```
person isa entity.
employee ako person.
applicant ako person.

name isa field;
  belongs-to-entity(person).

application isa entity.
applicant-field isa field;
  belongs-to-entity(application);
  has-value-type(applicant).
appliction-date isa field;
  belongs-to-entity(application).

residence-permit-application ako application.
```

The reference data model makes it possible to connect the entities in the different web services and describe how these are related to each other, by reference to the independent concepts in the reference model. Thus it becomes possible to see which IT systems have information about applicants, and what information each system has about these.

The entities and fields used in real-world applications are unlikely to correspond exactly with the reference data model, and so we need more than one way to connect these. We propose four different association types for the purpose:

- `represents`, meaning that the correspondence is perfect. The application entity/field represents the same thing as the reference data model concept.
- `specialization-of`, meaning that overlap is perfect, but the application concept is more restricted than the reference concept.

- `overlaps-with`, meaning that there is a common subset, but the overlap is not perfect. This association is a warning sign, suggesting that the reference model may be in need of extension.
- `related-to`, meaning that the correspondence is very loose, but that this was the best existing match at the moment. Every use of this association type is really an admission of failure, and represents a problem that needs to be addressed at some point in the future.

We include the last level above because real-world IT systems are usually very far from perfect, and a solution like this allows concepts to be documented in the system before they really fit in. Thus, one can document problem areas without having to fix them first, while still recording that they are problems.

A simple example might look as follows:

```
pr-employee represents(employee).
pr-employee-name represents(name).
```

### 2.1 Comparison with existing practice

While the reference data model may sound like what enterprise architects call a canonical data model (CDM), it is not intended to fulfill the same purpose. Canonical data models and associated formats are intended to simplify information architecture by requiring all systems to communicate using the same model and format [Hoof07].

In practice, however, this is difficult. Applications tend to serve specialized business needs in their particular process contexts, so that very often indvidual applications need data beyond what fits into a CDM. Since the CDM is maintained centrally via what in practice tends to be a slow, bureaucratic process, many projects find their progress obstructed. In essence, they face a choice between leaving out some information, using a different format, or unilaterally extending the CDM. The usual solution in practice is not using the CDM.

The reference data model, by contrast, is not necessarily used for interchange, but as a reference point to relate the data models and interchange formats used in the organization to one another. Thus, individual projects do not need to wait for the reference model to be updated, but can proceed immediately.

### 2.2 Tooling

Once step 1 is complete the organization has a very good picture of the internal IT systems and their data available in Topic Maps form. The map can be searched and visualized in a number of different ways, and even queried with structured queries like "which IT systems contain information about people?" There are a number of free open source and commercial Topic Maps tools on the market which can be used to maintain and work with this structure.

Since Topic Maps are an ISO standard with good interoperability between tools, this makes the solution both vendor-independent and platform-independent, allowing organizations to choose the tools they prefer, and change their minds later.

In stark contrast to existing ESBs and enterprise information integration tools, these tools are either free or relatively inexpensive. As enterprise architecture improvement is an inherently expensive process whose only justification is the savings it potentially generates, this is an important point, improving the likely return on the investment.

Further, existing tools for working with CDMs tend to use a proprietary UML-like language for modelling, and often export the models to W3C XSD for the associated wire format. There is therefore a greater tendency for vendor lock-in with these tools. In addition, UML and XSD are weaker tools for information modelling than Topic Maps are. UML, for example, does not treat properties as first-class citizens, severely limiting what can be said about them. Similarly, XSD has very weak support for relationships between different types, taking a very syntax-bound view of extensibility.

## 3 Step 2: Basic generic services

While the topic map created in step 1 has value in itself as a kind of structured documentation we wish to take it a step further and actually make use of the structure to build web services. The goal is to create a more loosely coupled architecture than what is usual.

To be able to support such services the topic map must be extended as follows:

- The different web services must be categorized. One key category is “lookup”. That is, services which given an identifier return information about the identified entity.
- The different XML formats returned by the various web services must also be identified.
- Web services which can translate between the different XML formats must be categorized as “translators”, and their input and output formats described.

This could be described as follows:

```
pr-employee-data isa web-service;
  http://internal/GetEmployeeData ;
  service-category(lookup);
  based-on(pr);
  can-return(pr-employee);
  can-return(pr-employee-name);
  output-format(pr-xml).

pr-xml isa xml-format.
foo-xml isa xml-format.

pr2foo isa web-service;
  http://internal/pr2foo ;
  service-category(translator);
  input-format(pr-xml);
  output-format(foo-xml).
```

Given this we can build a generic lookup service, which takes three parameters:

- An identifier,
- An entity type,
- An XML format.

The service looks up in the topic map which web services provide lookup for the given entity type, and then does the lookup. If the format provided by the web service is not the desired format, a translator service is used to convert the data to the correct format.

This makes it possible to change the internal architecture away from the traditional ad hoc structure where each client is tightly coupled with the services it's based on. Instead, there is a transition to a "mediator service", which passes requests around to the correct services. Thus, which systems provide lookup services for what entities in what formats can change, without client services needing to be aware of these changes.

Thus, a client system needing data about application 34234231 in the `foo-xml` format can just ask the mediator for this information, without needing to worry about how this is actually provided.

Historically there have been many attempts to build infrastructures where clients dynamically search for, discover, and bind to services providing the necessary functionality. To goal has always been more dynamic and flexible architectures. An early example of this was the CORBA services, such as the CORBA Trading Object Service, and later alternatives like UDDI have followed. However, these initiatives all failed.

A major problem for this type of service has always been that they were based on very weak data models. For example, UDDI only allows search by a single criterion at a time, and has a very limited set of possible criteria. Further, these systems have been intented for use in unlimited context (like the open internet), and have had no limitation on the type of service which can be discovered and bound to.

The scenario we describe is far more limited. First of all, the context is a single organization, not the open internet. Secondly, there is only a limited set of service types. And thirdly, we do not expend service providers and clients to act completely independently of each other, but rather for the organization to carefully set up the service registry so that all clients find suitable services.

Thus we believe that our approach has a significantly greater chance of success, through the more limited scope of the approach.

## 4 Step 3: Removing redundancy

Once steps 1 and 2 are completed the foundation is in place for handling some of the challenges identified in section 1. We can now easily identify cases where the same source data is maintained in different applications, and choose an application to be the master for this type of data. The other applications must then be rewritten as clients of the master, and web services must be provided so that the clients can access the data.

The web services must then be registered in the topic map so that the client applications can look up the necessary data via the mediator services. This must be a closely controlled process, so that clients always find the services and data they request. Obviously, this will be a slow and careful process, as the organization's internal IT systems are being rebuilt while running.

## 5 Step 4: Collecting reference data

In most organizations general reference data, like, say, lists of countries, will be maintained independently in different IT systems. This means that the work to update these lists must be replicated. It also means that there is every chance that the identifiers for these concepts in different IT systems will be different, thus making it difficult to integrate data from the different systems.

There is money to be saved, and data quality to be improved, by building a central reference data service to maintain this information. In addition, the potential for data integration can be greatly improved by ensuring that all applications use the same identifiers for the various entities.

The obvious way to do this is to maintain the reference data in the topic map, probably using the default Topic Maps editor, so that no extra applications need to be built to handle this. Further, PSIs should be used to identify the entities. The data might look as follows:

```
country isa entity;
  http://psi.oasis-open.org/iso/3166/#country .

norway isa country;
  - "Norge" @ norwegian;
  - "Norway";
  - official-name: "Kongeriket Norge" @ norwegian;
  - official-name: "Kingdom of Norway"
  http://psi.oasis-open.org/iso/3166/#578 .

# and so on \dots
```

In addition, a web service must be provided for downloading the full set of instances of a given entity type with their instance data (such as names). There must also be a web service providing a feed of the changes to entities of a given type. Thus other systems can get the initial list of entities, and maintain its of own list in sync with that from the central repository.

These web services will use the PSIs shown in the example to identify the entity types and their instances.

## 6 Step 5: Access control

Each organization has different needs for access control, but most do have rules for who is allowed to see and change what, and these requirements are often of great importance for the organization. At the same time, access control tends to be implemented ad hoc in each separate system, which means that there are great possibilities for inconsistencies and errors. It also very difficult to audit the access configurations and rules across the different systems to verify that these are correct.

The US Department of Energy, which is responsible for the nuclear power stations in the US, and which also produces nuclear weapons from the byproducts of power production, had this problem. The DoE had very extensive rules for which information

was restricted, and the degree of restriction. The rules were different for different sites around the US, which meant that each site had a Word document of a few hundred pages describing the rules. With a few hundred sites, the total information mass was a major challenge to handle.

Violations of these rules are punishable by law, in some cases even by death, and so it was of great importance for the rules to be consistent and correct across the US. To this end, the rules were taken out of the Word documents and represented in a topic map, in order to provide a better overview, and to make it easier to compare rules across different axes [Mason03].

We propose taking a similar approach to access control. Since the reference data model is already in the topic map it becomes possible to describe user groups in the topic map and to connect their privileges to different elements in the reference model. If some groups only have access within certain parts of the data, for example limited by country, this can also be expressed in the topic map, since it already contains the necessary reference data.

There is no need to maintain information about individual users and the user groups they belong to in the topic map, since the organization probably already has systems to handle this, like Microsoft Active Directory or something similar. The topic map only describes the access rights of each user group.

A web service must be provided so that each IT system can look up the rules from the topic map. We assume that individual IT systems will have to implement some of the access control logic individually for the foreseeable future, since this tends to be tightly integrated into the user interface of each application. Still, we wish to move as much as possible of this out into "the cloud".

The generic services, like the generic lookup service from step 2, can use the access control information in the topic map directly, and thus reduce the duplication of access control logic.

## 7 Step 6: Generic data searching

The generic lookup service can solve many data access needs, but realistically speaking it is so primitive that many needs will be beyond what it can handle. A more powerful solution is to give clients the ability to specify their requirements in more detail, preferably with both which fields to return and detailed criteria for which entities to return. To make a detailed API allowing this to be described programatically is unlikely to be easy or very usable if done. A far better solution is to use a query language.

We propose using SPARQL, a query language for RDF standardized by the World Wide Web Consortium [SPARQL]. RDF has a very simple data model, essentially just entities with fields and values, which matches our proposed reference data model very well. SPARQL can therefore be used to formulate searches in terms of the reference data model, and a new mediator service can therefore be built to pass queries to data sources based on its information about which sources contain what data.

A query for all countries from which residence permit applications were received in March 2010 might look as follows:

```
select ?country where
  { ?a rdf:type residence-permit-application .
    ?a applicant-field ?applicant .
    ?applicant nationality ?country .
    ?a application-date ?d .
    filter ( xsd:date(?d) < xsd:date("2010-04-01") ).
    filter ( xsd:date(?d) >= xsd:date("2010-03-01") ). }
```

Analyzing this query the mediator will see that it is for residence permit applications (`?a`), applicants (`?applicant`), dates (`?d`), and countries (`?country`), but first and foremost the applications. It will thus be able to look up the system which has information about these, and pass the query there. A wrapper around the system supporting SPARQL will have to be implemented for this to work.

As for the shape of the web service itself and the data format it returns for search results there is a separate standard specifying both, which can be used [SPARQL-PROTOCOL]

### 7.1 The choice of SPARQL

The choice of SPARQL, an RDF query language, for a Topic Maps-based solution may seem strange at first sight. Other alternatives could be tolog [Garshol05] or the coming TMQL. However, SPARQL has a number of advantages over these:

- SPARQL is a standard, which tolog is not.
- SPARQL exists now, which TMQL does not.
- SPARQL has a deliberately limited feature set, designed to allow query federation, and is easier to implement in a wrapper system.

Another reason for the choice of SPARQL is that, at least initially, very few of the systems being queried are Topic Maps-based, but are instead wrapped legacy systems. Thus, whether the query language used is RDF-based or Topic Maps-based is of less importance. However, even Topic Maps-based systems can be queried with SPARQL using TMSPARQL [Ahmed09].

Overall, SPARQL therefore seems to be the best choice.

## 8 Step 7: Semantic formats on the wire

Eventually, systems should start using semantic formats like XTM directly for interchange, using the reference data model directly to represent data in XTM. In a way, this means returning to the concept of a canonical data model, but with semantic technologies it can be made into less of a straightjacket.

For example, if the application needs to include a field which is not part of the CDM it can simply create a new PSI for the field and use that. In a format like XTM, the presence of an extra field causes no problems for consumers.

The field should be anchored to the reference data model as far as possible via subtyping from an existing field in cases where this is semantically correct. Subtyping

makes it possible to find the field value both via the supertype and via the subtype, making the new subtype effectively a more specialized alias for the old supertype.

If there is no suitable field to subtype from one can still start using the field right away, if necessary. Later, once the request for a new field has been accepted and the reference model extended, a subtyping statement can be added to connect the original specialized field with the new field in the reference model.

## 9 Related work

A related approach to enterprise information integration is described in [Borge10], which describes the use of Topic Maps to implement a kind of data warehousing. The benefit of this approach is that it provides a simple and lightweight implementation of data integration which can support a wide variety of use cases. It meets some of the same requirements as the proposals in this paper, but is insufficient as a means to rearchitect an enterprise's information architecture. (It should be added that this was never the intention.) Overall, the approach in [Borge10] is complementary to the approach proposed here, and the two could easily and beneficially be integrated into a single approach.

Extensive work is in progress in the research community on what is termed "semantic web services" [Wikipedia]. This work primarily builds on the OWL ontology description language and the capabilities this provides for reasoning over ontology descriptions. OWL has sophisticated mechanisms for describing the relationships between parts of ontologies that go far beyond what is outlined in section 2. However, these initiatives, while interesting and worthwhile, are not very closely connected to the requirements of enterprises. Thus, it is difficult to see how they can be applied to the kind of challenges we have described here.

It should be added that while OWL is far more powerful than the approach we described in step 2, OWL is also a logic language. It is essentially an RDF representation of Description Logic notation, which gives it some useful, mathematically proven, properties. However, it also makes it very difficult for non-specialists to understand, which is a real challenge given that enterprises are meant to maintain these models themselves. For this reason we have advocated a more limited approach in this paper.

## 10 Conclusion

We believe the approach advocated in this paper can help organizations struggling with their IT infrastructure considerably, but must confess that we have not yet implemented it in any organization. It therefore remains to be proven that the approach really does provide benefits in real life use. We are currently engaged in discussions with a number of organizations struggling with the issues described in this paper, and given the level of interest from these, we expect to be able to put these ideas into practice in the near future.

## Acknowledgements

The ideas presented here owe their beginnings and inspiration to a project initiated by André Torkveen and Tore Bjerke. They have been further developed in discussions with Stian Danenbarger, Graham Moore, and Axel Borge.

Input from Simen Sommerfeldt on the state of the art in enterprise architecture greatly improved the paper.

Thanks also to the anonymous reviewers for their very useful comments.

## References

Borge10. A. Borge; Topic Maps as key enabler for common archive functionality across a range of IT-systems; proceedings of TMRA 2010 (this volume)

Brady04. T. Brady; Representing Software System Information in a Topic Map: A markup-centered approach; Proceedings of Extreme Markup 2004; IDEAlliance; Montréal, Canada, 2004; http://conferences.idealliance.org/extreme/html/2004/Brady01/EML2004Brady01.html

Economist10. Computer says no; The Economist, 2010-07-22. http://www.economist.com/node/16646044?story_id=16646044

Gulbrandsen03. A. D. Gulbrandsen; Prosjekt Houston: En kunnskapsbase for driftsdokumentasjon; Emnekart Norge 2003; Oslo, Norge; 2003-11-26; http://www.emnekart.no/2003/are-gulbrandsen.pdf

Mason03. J. D. Mason; Topic Maps for Managing Classification Guidance; Proceedings of Extreme Markup 2003; IDEAlliance; Montréal, Canada, 2003; http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.139.3660&rep=rep1&type=pdf

SPARQL. E. Prud'hommeaux, A. Seaborne; SPARQL Query Language for RDF; W3C Recommendation 15 January 2008; World Wide Web Consortium; http://www.w3.org/TR/rdf-sparql-query/

SPARQL-PROTOCOL. K. G. Clark; L. Feigenbaum; E. Torres; SPARQL Query Language for RDF; W3C Recommendation 15 January 2008; World Wide Web Consortium; http://www.w3.org/TR/rdf-sparql-protocol/

Hoof07. J. van Hoof, How to mediate semantics in an EDA, blog post on "SOA and EDA" blog, 2007-04-19, http://soa-eda.blogspot.com/2007/04/how-to-mediate-semantics-in-eda.html

Wikipedia. Semantic Web Services, Wikipedia article, 2010-08-27 revision, http://en.wikipedia.org/wiki/Semantic_Web_Services

McKendrick08. J. McKendrick; Enterprise Service Busted?; ZDNet; 2008-07-22; http://www.zdnet.com/blog/service-oriented/enterprise-service-busted/1149

Hinchcliffe09. D. Hinchcliffe; Fixing Enterprise Architecture: Balancing the Forces of Change in the Modern Organization; ebizq.net; 2009-09-03; http://www.ebizq.net/blogs/enterprise/2009/09/fixing_enterprise_architecture.php

Ahmed09. K. Ahmed; Making Topic Maps SPARQL; In: Maicher, L.; Garshol, L. M.: Linked Topic Maps, Leipziger Beiträge zur Informatik: Band XIX; Leipzig Universität Verlag; 2009. http://www.topicmapslab.de/publications/making_topic_maps_sparql

Garshol05. L. M. Garshol; tolog – A Topic Maps Query Language; In: Maicher, L.; Garshol, L. M.: Charting the Topic Maps Research and Applications Landscape, TMRA 2005. LNAI 3873, Springer, (2006); http://www.garshol.priv.no/download/text/tolog.pdf

# Live Integration Framework

Christian Haß and Sven Krosse

University of Leipzig, Johannisgasse 26, 04103 Leipzig, Germany
\{hass,krosse\}@informatik.uni-leipzig.de

**Abstract.** In the last years the concept of open and linked data gained more and more popularity, but still most application store their data within different and heterogeneous models. By defining an integration framework, information from different sources can be represented in the same format, for instance as a topic map. In combination with the concept of virtual merging, the integration layer can then provide a new view for remote data without any redundancy and caching. All information about real world subjects could be concentrate in one data hub.

## 1 Introduction

In the last years the concept of open data and linked data became more popular. Applications and web services became part of the data cloud and provide information and knowledge in different data formats like XML, database snippets or even Excel[1] spread sheets. All these resource types are based on different model paradigms or knowledge representations and aggravate the concept of linked data. Information items of different linked data sources may be duplicated which leads to redundancy.

It is necessary to use information resources within one paradigm to enable logical and non-redundant conclusions. Each data set representing the same subject should be handled as equal and all information about the subject should be concentrated within one data item, acting as a knowledge hub. The model of Topic Maps offers a powerful opportunity to realize this in a quite easily way. Even if more than one subject uses the same identity, it will be merged by the used Topic Maps engine. As we know, most of the provided information are not modelled as Topic Maps, but isn't it possible to convert them to behave like Topic Maps?

Topic Maps can be used to create a reduced view on data located in different stores. By using the concept of virtual merging, a common view onto the data can be created.

## 2 Storage Technologies

In the sector of information representation and knowledge management a huge number of different data formats have been established. Data may be available as public resources (as a result of the open and linked data initiative) or can be located in data silos disconnected from the open web. Because of isolated information domains, it is not possible to generate additional knowledge through combining different data sources.

[1] Excel and Microsoft Office are registered trademarks of Microsoft

In the business sector, information often is stored by one of the *big three* (relational databases, XML-based file formats or Excel files). Of course there are also other formats, but they are often unstructured or cannot be used in the way, we want them to.

### 2.1 Relational Databases

Relational databases are the most commonly used storage technology. They allow storage of and high-performance access to large amounts of information. Relational databases are based on the relational database-model described by Edgar F. Codd in 1970 [1].

**Structure of Relational Databases** Information is stored within relations represented as tables. Each table has a unique name and consists of a finite number of tuples of the same type (represented by the rows of the table). Each tuple consist of a finite number of attributes (represented by the columns of the table).

The tables and related columns are defined by the database schema. The schema is completely generic and can be freely defined by the user.

To identify a specific tuple, a key is used. A key is the value of one or more tuple attributes and has to be unique for each entry in the specific table.

By definition, all attributes of a tuple are in relation to each other but it is also possible to create relations between two or more tuples. This can be realized by creating a relation between their keys. A simple one to one relation can be defined directly by adding a foreign key attribute to a tuple while n-nary relations need to be defined in additional tables.

**Information Access** The most common way to access information stored in a relational database is via the Structured Query Language (SQL). SQL is not limited to retrieving information from the database, it also is able to modify, delete or insert information as well as to change the database schema.

### 2.2 XML-based Contents

The Extensible Markup Language (XML) [2] is a machine-readable representation for structured data. XML was designed as a simple and platform-independent format. On the top of the XML specification, other data formats were established like RSS or XHTML. Because of that, it is possible to handle each of these formats in a common way.

**Structure of an XML Document** An XML document represents information as a tree structure of information objects (nodes, attributes and text content). As a strictly defined dataformat, information objects may not overlap and represent an enclosed out-cut of the described domain. An XML node can be child of another node, representing a more general term. Additionally, a node can contain an unspecific number of attributes represented by key-value pairs. XML nodes which are leafs of the document tree, often contain simple text content.

**Identity of Information Objects** XML nodes or attributes can be identified by their relative or absolute position within the document tree. The path can be described by an XPath [3] statement. In addition, if an XML node contains an attribute of type *ID*, this can be used as a unique identification in the context of the document. Attributes of a specific node need to have a unique key.

### 2.3 Microsoft Excel

Microsoft Excel provides a semi-structured file format. Information is stored within different entities, represented by sheets that are similar to tables (in relational databases) and have mandatory unique names in the context of the whole document. In contrast to relational databases, a sheet can represent tuples of different types, which makes it more complicated to address a specific tuple. A sheet is also organized as a two dimensional matrix of columns and rows. Columns are named by lexical and rows by numerical identities. Because of that, each data cell can be addressed by the triple of sheet name, column and row identity.

**Office Open XML file formats** Since 2007, the Microsoft Office suite stores Excel sheets as structured files similar to XML. The structure of such files is restricted by the ISO standard 29500 [4]. The benefit of the new office file format for Excel sheets is the possibility to use equivalent access mechanisms for XML and Excel.

## 3 Design a Mapping Framework

In the context of this document, only the mapping of relational databases will be discussed in detail. In the scope of this document, the term mapping means the matching between information stored in different data models. More specifically, the matching between the data model of the store that is to be integrated (relational database, XML file, etc.) and the Topic Maps Data Model(TMDM) [5].

Live integration means, that no information is copied or cached. Each Topic Map construct, representing a specific information, will be generated at the very moment the information is accessed. If a mapped value changes in the integrated store, the value of the related Topic Maps construct changes as well.

**Requirements** To map information to the TMDM, it is necessary to define or at least generate one identifier for each subject. This is especially important if information from different data stores should be integrated into a simple topic map. The identifier has to explicitly represent the specific subject in all integrated stores, to allow merging of same subjects but also to avoid merging of subjects which are not the same. Most data stores use identities which address subjects in the specific store and cannot be used as a global identifier. Because of this, it may be necessary to add additional information to the store in order to create representative identifiers.

In certain cases it is requested or even necessary to add ontology information to the mapped constructs. For instance, some Topic Maps constructs need to have types.

Therefore, the mapping has to provide the possibility to define the ontology. However, it is sufficient to specify only an identifier for each ontology topic. Due to the concept of virtual merging, additional information can be merged in, even if the integrated store does not provide write access.

**Restrictions** As already mentioned in the previous section, there are certain restrictions with respect to live integrating topic maps. Each store needs a predefined mapping which is static during run-time. This means that all aspects related to ontology that are modeled in the mapping are also static. Additionally, it is also not possible to create new constructs, since they may need additional mapping information. Because of that, the designed framework will only support read access in the first iteration. However, it may be possible to provide at least limited write access in a later version.

### 3.1 Mapping Relational Databases

As shown in 2.1, relational databases are composed of a number of tables which contain a number of tuples of the same type. Information stored in one tuple has to be mapped to sub-constructs of exactly one topic. This is necessary because all tuple attributes are identified by the same key. Thus, the table represents the topic type and each tuple an instance of this type.

**Identifier** The keys which are used to identify the specific tuple can be used to generate item identifiers, since they have to be unique for a specific table. The identifier could be generated from the Topic Map base locator in combination with the table and column name matching the following schema.

```
<Topic Map Base Locator>/<Table Name>/<Column Name>/<Key Value>
```

As discussed in section 3 the mapping of suitable subject identifiers or locators needs identities which are representative for the specific topic. This could be a customer or product ID but also URLs like web-links or email addresses. To generate valid identifiers from those identities it may be necessary to define a prefix.

```
<Prefix><Identity Value>
```

**Characteristics and Associations** All attribute values can be converted to strings so that it is possible to map all attribute fields to a name or an occurrence respectively. However, it makes more sense to map values which are not strings to constructs which can distinguish between different data types. The data type could be defined in the mapping or be directly retrieved from the database schema.

Theoretically, it is possible to map certain attributes to name variants, but the relational database model does not define attributes which are only valid in the scope of another attribute. Because of that, the mapping of variants is not further discussed in this paper.

Associations can be mapped from tuples which contain more than one key attribute. The association mapping needs to specify the association type and a number of roles. Each role has to specify the role type and the attribute that contains the player key.

**Scopes and Reification** The concept of scopes is not defined in the relational database model. However, it could be required to define static scopes for certain constructs. Therefore, it would be a good approach to allow a scope definition in the mapping.

If the tuple originally representing the association contains additional information which is not already mapped to a topic, this information can be mapped to names and occurrences of the reifier of this association.

## 4 Prototypical Implementation

To show that the general concept of a live integration framework is feasible, a prototype which allows the integration of relational databases was developed. The prototype is implemented as a MaJorToM [7] store. MaJorToM is a Topic Maps engine which uses lightweight objects to represent the different Topic Map construct. These constructs are simple stubs which hold no specific information. The actual information is held by an underlying store and can be requested in context of the specific objects. This makes MaJorToM an ideal engine to realize a live integration framework, since no information needs to be stored in the Topic Map constructs. The current prototype only supports tuple keys from single attributes. This restriction was made to simplify the development and will be removed in the final framework.

### 4.1 Topic Mapping

As described in 3.1, a topic represents one specific database tuple. The attributes of the tuple can be represented by the topic names, occurrences or identifiers. Thus, the information, in which table and column a specific value can be found, needs to be defined in the related mapping definitions. Additionally the information, which column holds the value, is needed. The mapping is specified in an XML file. For each topic type a topic mapping node defines the related name, occurrence and identifier mappings as sub-nodes. Each topic mapping needs an ID to make it reference-able in a role mapping. The mapping of one attribute is defined by the combination of table, column and key column name.

```
<link table="<table name>" column="<attribute column name>"
keyColumn="<key column name>"/>
```

A topic is initially accessed, the first time it is necessary to select the related key and store it in context to the topic object. The key can be retrieved via the identifier mapping, since an identifier value as well has to be unique for a specific table. Because of that, each topic mapping needs to specify at least one identifier mapping. The identifier mapping can be used to generate the following SQL query to retrieve the key.

```
SELECT <key column name> FROM <table name> WHERE
<attribute column name>='<identifier reference>'
```

Once the key is known, all mapped values can be retrieved by the following query:

```
SELECT <attribute column name> FROM <table name> WHERE
<key column name>='<key>'
```

Instead of the specification of table and column names, a complete query could be defined for name or occurrence attributes. This will not be applicable for identifier mappings since, as already discussed, those mappings are also used to retrieve the key values. A query specified in this way needs to return exactly one value (or NULL) and has to have exactly one parameter which will be replaced by the specific key value. Since relational databases offer a large amount of powerful functions, names and occurrences which do not exist in the database this way could be mapped. The following example shows, how a new name construct could be mapped by the combination of two columns:

```
<link query="
SELECT CONCAT(<name column>," ",<surname column>)
FROM <table name>
WHERE <key column>='?'"
/>
```

### 4.2 Association Mapping

An association can be mapped from tuples which have more than one key attribute. The identification of a specific association is possible by the combination of all key values related to their columns. If more than one tuple exists for a specific combination, e.g. if the table has an additional key column which is irrelevant for the association, each tuple will represent the same association.

An association mapping specifies the table name and a finite number of role mappings. Each role mapping has to specify the role type and the ID of the related topic mapping which acts as the role player. This is necessary to check if the specific topic actually exists. Additionally, it needs to specify the column name in which the topic key can be found. The information from all role mappings can be used to formulate a query, to check, if a specific association exists:

```
SELECT * FROM <table name>
WHERE <key role 1>="<key 1>" AND <key role 2>="<key 2>"
AND \dots AND <key role n>="<key n>"
```

The same information can also be used to select all associations of a specific association mapping.

```
SELECT <key role 1>,<key role 2>,\dots,<key role n>
FROM <table name>
```

A valid association is defined by a result which has no NULL values under the condition that all related role players currently exist.

### 4.3 Example

In this example we have two databases in which customer related information is stored.

Figure 1 shows the customer table of the first database (A) which only has one entry. Figure 2 shows the product and order table of the same database. Figure 3 describes the customer table in the second database (B) which holds additional information of the same customer, i.e. the customer with the same *customer id*.

The mapping for database A follows:

| customer | | |
|---|---|---|
| customer_id | name | address |
| 1 | A Dummy | Street Nr 1, 12345 Town |

**Fig. 1.** Customer Table in Database A

| products | | | |
|---|---|---|---|
| product_id | name | description | price |
| 1 | Product One | Example | 1,00 € |
| 2 | Product Two | Example | 2,00 € |

| order | | |
|---|---|---|
| product_id | customer_id | date |
| 2 | 1 | 01.01.10 |

**Fig. 2.** Product and Order Table in Database A

| customer | | | | |
|---|---|---|---|---|
| customer_id | name | surname | email | phone |
| 1 | A | Dummy | a@dummy.de | 123456789 |

**Fig. 3.** Customer Table in Database B

```
<jli-mapping
  xmlns="http://www.topicmapslab.de/mapping"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
   "http://www.topicmapslab.de/mapping mapping.xsd">
  <connection-settings>
    <url>//database.a:3306</url>
    <name>example</name> <user>user</user>
    <password>password</password>
    <catalog>public</catalog>
  </connection-settings>

  <mapping>
    <topic id="customer">
      <type>http://example/customer</type>
      <identifier type="SI" base="http://example/customer/">
        <jdbc_link table="customer" column="customer_id"/>
      </identifier>
      <name type="http://example/fullname">
        <jdbc_link table="customer" column="name"
         keyColumn="customer_id"/>
      </name>
      <occurrence type="http://example/address"
       datatype="xsd:string">
        <jdbc_link table="customer" column="address"
         keyColumn="customer_id"/>
      </occurrence>
    </topic>
```

```
    <topic id="product">
      <type>http://example/product</type>
      <identifier type="SI" base="http://example/product/">
        <jdbc_link table="product" column="product_id"/>
      </identifier>
      <name>
        <jdbc_link table="product" column="name"
         keyColumn="product_id"/>
      </name>
      <occurrence type="http://example/discription"
       datatype="xsd:string">
        <jdbc_link table="product" column="description"
         keyColumn="product_id"/>
      </occurrence>
      <occurrence type="http://example/price"
       datatype="xsd:string">
        <jdbc_link table="product" column="price"
         keyColumn="product_id"/>
      </occurrence>
    </topic>
    <association type="http://example/oder" table="order">
      <role topicId="customer" type="http://example/ordered">
        <jdbc_link column="customer_id"/>
      </role>
      <role topicId="product" type="http://example/was_ordered">
        <jdbc_link column="product_id"/>
      </role>
      <reifier>
        <occurrence type="http://example/order_date"
         datatype="xsd:date">
          <jdbc_link table="order" column="date"/>
        </occurrence>
      </reifier>
    </association>
  </mapping>
</jli-mapping>
```

In the mapping, two topic types are defined, one for the customer table and one for the product table. Additionally, an association based on the order table is defined. This mapping file in combination with database A results in the following topic map described in CTM [6].

```
%encoding "UTF-8"
%version 1.0

<http://example/customer/1> isa <http://example/customer>;
    - <http://mycompany/fullname>:"A Dummy";
    <http://mycompany/address>:"Street Nr 1, 12345 Town".

<http://example/product/1> isa <http://example/product>;
    - "Product One";
```

```
    <http://example/discription>:"Example";
    <http://example/price>:"1,00€".

<http://example/product/2> isa <http://example/product>;
    - "Product Two";
    <http://example/discription>:"Example";
    <http://example/price>:"2,00€".

<http://example/oder>(
 <http://example/ordered> : <http://example/customer/1>,
 <http://example/was_ordered> : <http://example/product/2>)
 ~[<http://example/order_date> :
   "01.01.10"^^http://www.w3.org/2001/XMLSchema#date]
```

In the following mapping for database B, the same way to generate the subject identifier for the customer topics is used, i.e. the same *base value*.

```
<jli-mapping
  xmlns="http://www.topicmapslab.de/mapping"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
   "http://www.topicmapslab.de/mapping mapping.xsd">
  <connection-settings> <url>//database.b:3306</url>
    <name>example</name> <user>user</user>
    <password>password</password>
    <catalog>public</catalog>
  </connection-settings>
  <mapping>
    <topic id="customer">
      <type>http://example/customer</type>
      <identifier type="SI" base="http://example/customer/">
        <jdbc_link table="customer" column="customer_id"/>
      </identifier>
      <name type="http://example/name">
        <jdbc_link table="customer" column="name"
         keyColumn="customer_id"/>
      </name>
      <name type="http://example/surname">
        <jdbc_link table="customer" column="surname"
         keyColumn="customer_id"/>
      </name>
      <occurrence type="http://example/email"
       datatype="xsd:string">
        <jdbc_link table="customer" column="email"
         keyColumn="customer_id"/>
      </occurrence>
       <occurrence type="http://example/phone"
        datatype="xsd:integer">
        <jdbc_link table="customer" column="phone"
         keyColumn="customer_id"/>
      </occurrence>
```

```
    </topic>
  </mapping>
</jli-mapping>
```

The resulting topic map will look like this:

```
%encoding "UTF-8"
%version 1.0

<http://example/customer/1> isa <http://example/customer>;
    - <http://mycompany/name>:"A";
    - <http://mycompany/surname>:"Dummy";
    <http://mycompany/email>:"a@dummmy.de";
    <http://example/phone>:
         "123456789"^^http://www.w3.org/2001/XMLSchema#integer.
```

Since there are customers from both databases, that have identitcal *customer ids* and thus identical subject identifiers, it is possible to merge this information and generate a common view.

## 5 Benefits of an Integration Framework

The described framework is designed as a lightweight abstraction layer between the real data base and the Topic Maps based application. Information that is not modeled as a Topic Map can be represented as such. Due to the concept of merging, it is possible to link information from heterogeneous data sources. Knowledge which is currently isolated in data silos can be made available in a centralized information hub without loosing knowledge about origin and ownership. To realize this, it is not necessary to change the existing infrastructure or any work flow. The integration framework can be used without any influence on existing systems or applications.

## 6 Related Work

The integration of different data sources using Topic Maps based applications is adressed by several studies and prototypical projects in the past. In [8], which focused on a similar approach, the authors offer a prototypical implementation of an integration framework for relational data sources using a CTM based mapping definition which is limited to the most common combinations and a specific Topic Maps Engine [9].

A middleware framework for accessing knowledge based on Topic Maps technology [10] or a protocol for remote access of Topic Maps [11] are only some of the other solutions.

## 7 Outlook

The paper describes a general integration concept as well as a prototypical implementation of an RDBMS integration framework. In the next iteration the integration of XML

and Excel documents will be implemented as discussed in section 2. Additionally, it will also be possible to integrate SAP systems. The current concept is restricted to a read-only integration layer, however it will be possible to allow at least limited write access, for instance the modification of literals. Depending on real world scenarios, the framework could be extended to support dynamic scoping or name variants.

## References

1. Codd, E. F.: A Relational Model of Data for Large Shared Data Banks In: Communications of the ACM vol.13, p. 377–387, 1970
2. W3C Recoomendation:Extensuble Markup Language (XML) 1.0, Fifth Edition, 2008-11-26
3. W3C Recoomendation:XML Path Language (XPath) 2.0, 2007-01-23
4. ISO/IEC 29500: Information technology – Document description and processing languages – Office Open XML file formats, 2008-11-18
5. ISO/IEC IS 13250-2: Information Technology – Document Description and Processing Languages – Topic Maps – Data Model. International Organization for Standardization, Geneva, Switzerland, 2008-06-03 http://www.isotopicmaps.org/sam/sam-model/
6. ISO/IEC 13250-6: Topic Maps — Compact Syntax. Current CTM draft (2010-03-31) http://www.isotopicmaps.org/ctm/ctm.html
7. MaJorToM Google Code Project: MaJorToM – Merging Java Topic Maps engine http://code.google.com/p/majortom/
8. Neidhart, T, Pinchuk, R., Valentin B.: Semantic Integration of Relation Data Sources With Topic Maps In: Maicher, L.; Garshol, L. M.: Linked Topic Maps pp. 185–192 Springer, Berlin (2009)
9. TopiEngine: An Open Source Topic Maps Engine http://launchpad.net/topiengine
10. Barta, R.: Knowledge-Oriented Middleware Using Topic Maps. In: Maicher, L.; Garshol, L. M.: Scaling Topic Maps pp. 98–115 Springer, Berlin (2007)
11. Garshol, L., M.: TMRAP – Topic Maps Remote Access Protocol In: Maicher, L.; Park, J.: Charting the Topic Maps Research and Applications Landscape pp. 53–68 Springer, Berlin (2005)

# Part VII

# Theoretical Topic Maps Research

# Topic Merge Scenarios for Knowledge Federation

Jack Park

Knowledge Media Institute, Open University, UK
jackpark@gmail.com

**Abstract.** Climate change is a growing concern to humankind, since the dominant view argues for rapid, significant changes in human behavior to avert catastrophic consequences. This is a complex problem, known as a wicked problem. A productive way forward is through creative, critical dialogue. Such dialogue requires new kinds of socio-technical infrastructure. We offer a socio-technical infrastructure, described as a boundary infrastructure, based on improvements to existing and emerging Issue-based Information Systems (IBIS) conversation platforms. IBIS is an emerging *lingua franca* of structured discourse. We report on a core function in that boundary infrastructure: the topic merge architectures of topic mapping frameworks. We use two scenarios known to exist in our research platform to develop a novel merge architecture that supports *virtual merges* of topics.

## 1 Background

Climate change is a growing concern to humankind, since the dominant view argues for rapid, significant changes in human behavior to avert catastrophic consequences. Those changes are proposed against a backdrop of lifestyle and economic change due to proposed and emerging mitigation plans. This is a complex problem, known as a wicked problem (Conklin, 2005). A productive way forward is through creative, critical dialogue. Such dialogue requires new kinds of socio-technical infrastructure. As part of a research and development program aimed at a prototype of a new kind of socio-technical infrastructure, we are developing an open source collective sensemaking platform we call Bloomer[1]. Bloomer serves two purposes in our research. One purpose is to support a thesis project that explores the ability to *federate* structured conversations (Park, 2010); the other purpose is to provide a socio-technical infrastructure, a boundary infrastructure (Bowker & Star, 1999) for collective intelligence, described as a *knowledge garden* (Park, 2008). From (Bowker & Star, 1999, p.313):

> "Any working infrastructure serves multiple communities of practice simultaneously be these within a single organization or distributed across multiple organizations. . . .Boundary infrastructures by and large do the work that is required to keep things moving along. Because they deal in regimes and networks of boundary objects (and not of unitary, well-defined objects), boundary infrastructures have sufficient play to allow for location variation together with sufficient consistent structure to allow for the full array of bureaucratic tools (forms, statistics, and so forth) to be applied."

[1] Bloomer: http://code.google.com/p/bloomer/

Bloomer combines a variety of web-based portals, including MediaWiki[2], Drupal[3], Cohere[4], and others, each communicating with a topic map platform we call TopicSpaces. To MediaWiki, we added a new extension that enables IBIS conversations to be conducted in the wiki. From this combination of platforms and from our goal to federate information resources, we derive novel topic merging situations, two of which we describe in this report.

Our use of the term *federate* entails topic maps. We posit that subject-centric merging of topics captured in social sensemaking settings offers opportunities for collaboration based on participants' discovery of like-minded others. Federation, in our sense of the word, is both a noun and serves as an act. The federation act is that of topic maps merging processes; a federation, the noun, is a collection of people, information resources, and the boundary infrastructure together with its many related boundary objects (Star, 1989). A topic in a topic map, like a concept drawn on a chalk board, is an instance of a boundary object; it is co-owned by all participants and serves as a place where resources are shared.

We describe our research in relation to two scenarios that arise in the Bloomer platform. When one creates a federation that facilitates social contributions from many participants on varieties of user interface platforms, one of our scenarios emerges: Federating representations of human participants – humans as topics. As we shall soon see, identifiers of the same individual can present to the federation as distinctly different depending on the platform used. Our second scenario is precisely that which is the subject of our thesis research: federation of structured discourse (Park, 2010). We describe these scenarios in more detail below after a brief review of merge technology issues.

In the following, we will describe concepts of merging of topics. Our research supports two kinds of merge actions: those which are software agent-based, guided by rules that inspect and vote for or against a merge between two topics, and those which are the result of human direction. Topic merging entails a single consideration when comparing two representations – called *subject proxies*: *do these two subject proxies represent the same subject*? That single question raises opportunities for research and innovation. In the remaining parts of this report, we examine a popular implementation of topic map merging technology, compare that to perceptions of the federation process that require merging, and then describe a new approach to implementation of merge processes we call *virtual merging*. We then describe two merge scenarios appropriate to the Bloomer project, its mission, and our thesis research.

## 2 Topic Maps Technologies

For this research, we identify two distinct approaches (among possibly many) to the fabrication of topic map platforms. A popular platform is based on the XML topic maps standard, two implementations of which are known by the acronyms XTM[5] (Pepper & Moore, 2001) and TMDM[6] (Garshol & Moore, 2008). Another implementation approach

[2] MediaWiki: http://www.mediawiki.org/
[3] Drupal: http://drupal.org/
[4] Cohere: http://cohere.open.ac.uk/
[5] XTM: XML Topic Maps
[6] TMDM: Topic Maps Data Model

is known as by its acronym TMRM[7] (Durusau et al., 2007). We explore differences in the two approaches to topic maps architectures distinguished by the XML and TMRM approaches. In the following, the term *locator* refers to an identifier (string) for an object that is unique to the database in which the object is stored.

In XML topic maps, three primary objects exist: *topics*, *associations*, and *occurrences*. A topic map is a collection of topic and association objects. Topics serve as containers for occurrence objects.

With TMRM, a topic map is a collection of subject proxies, where a subject proxy, like a topic object in XML topic maps, serves the purpose of containing representations of the subject for which it stands. Thus, the term *subject proxy* is an analog for a *topic*. The difference is this: a subject proxy is a container for property objects, subject properties. With the TMRM, a map's author may create property types – known as *keys*—to suit particular needs. In XML topic maps, a prescribed choice of property types is available, while others can be fabricated through the use of occurrence objects. In our view, both approaches serve the same purposes.

TMRM implementations do not distinguish between the concept types *topics*, *occurrences*, and *associations*; every concept type is always a subject proxy – a topic. Every topic is represented by collections of subject property objects – key-value pairs. In the TMRM, keys – property types – are defined as topics in the map; the specific locators of such topics serve as *keys* in property objects. While there are other means to accomplish the same ends, the overriding requirement of the TMRM is that the definition of those keys required for subject identification be recorded in a public document known as a *legend*.

TMRM specifies that, if there is any relationship (association) asserted between two topics, the association type is a defined topic, and the instance of that association linked between two *actor* topics is, itself, a subject proxy. Roles played by each actor are also topics in the map. Occurrences are topics that represent instances of *things* which *occur* 'out there' – recall, a map is a representation of some territory; occurrences are representations of topics in that territory. A particular web page, for instance, is represented by a subject proxy in a map.

In a TMRM implementation, we have the opportunity to model an XML topic map by simply declaring property types that mimic those declared by either XTM or TMDM. Doing so allows us to inherit useful merging algorithms found in existing platforms. At the same time, the TMRM allows us to explore other merge architectures. The opportunity to explore merge architectures animates this research. We articulate this research through the following desiderata:

- Provenance – used here to mean identification of sources of resources accumulated in any representation of any subject – is of sufficient importance that merge processes that are loss-free in terms of provenance maintenance are important
- Contested merges are those combinations of topic representations that may, at some later time, become suspect; at issue are two aspects of the merge process:
    - Merge accountability – records, histories of merges

[7] TMRM: Topic Maps Reference Model

- "Unwindability" – the ability to "un-merge", to unwind a merge and return the representations to their original forms.

That short list of issues leads to this proposition: it should be possible to perform a topic merge in such a way that:

- Provenance is fully maintained for every resource engaged in a topic merge
- All merges are accountable in terms of reasons offered for merge decisions
- All merges are contestable
- Any merge can be unwound, complete with reasons given for such decisions

Merge processes ask questions of subject identity. A trivial and rarely accurate form of subject identification lies in names for things. This author's name, when entered into a web query, is highly ambiguous – many *hits* occur, few of which are correct. But, names for things offer hints; combine hints with other properties such as roles played and a search returns greatly refined results. Combine this author's name with the term "topic map" and web query returns accurate results. Using that observation as a point of departure, we can compare the XML and TMRM architectures in terms of merging.

### 2.1 Scenarios

We identify two scenarios that we are experiencing in our work with the Bloomer project. One is an artifact of federating different platforms – specifically author identity; the other relates precisely to our thesis research: federating structured conversations. Specifically, the first scenario involves duplication of topics that represent actual human participants in sensemaking activities at different portals when introduced to the federation server, TopicSpaces. The second scenario involves federation of structured conversations accumulated at various portals. We introduce each next.

**Federating Participants** This federation issue is best described by our first encounter with the event. Consider that each participant needs an identity at each physical location where the participant either participates, or is modeled – represented – in a topic map. Consider the case of this author: logged in at a TopicSpaces federation server, identified there by the unique login name used. Same author, logged in at a MediaWiki instance, participating in a structured conversation which is being sent to the federation server. In the wiki, the very same individual logs in with the very same login name. The wiki, however, exhibits a quirky behavior: it capitalizes the first letter of the login name and transmits that to the federation server, where that author identity is not recognized. Thus, the scenario...

We have the same individual using the same identifier, but that identifier is mutated through a software artifact based on a particular wiki behavior. The federation server is programmed to deal with unknown authors by creating a new subject proxy for those that are not known to it. Since that method is algorithmic, when that same unknown individual enters the server in a different structure, the algorithm now knows that subject. Still, we experience the fact that we now have two different representations – subject proxies – for the very same subject. This situation would be hard to detect with algorithmic merge

detection processes; it does, however, submit to relatively easy detection when suspected, availing the opportunity for manual merge commands from appropriately credentialed administrators.

**Federating Structured Conversations** Our use of the term *structured conversation* refers specifically to the Issue-based Information Systems (IBIS) approach (Rittel & Webber, 1973; Conklin et al., 2003; Conklin, 2005). IBIS conversations are found under the names *dialogue map, issue map,* and *argument map*. In this approach, a *graph* represents a conversation in the form of nodes, which represent questions, answers, or arguments, are coupled together with labeled arcs. Figure 1 is an example of an IBIS *issue map* created using Compendium[8] (arc labels not shown).

When we federate conversations, we seek to provide views into conversations that avoid duplicate nodes. Thus, we merge conversations, node-by-node as described below. To introduce our problem space, consider two trivial IBIS conversations that, to native English speakers, appear to begin with the same question. Figure 2 presents those two conversations together.

Readers will recognize "carbon dioxide" and "co2" as names for the same gas. When those two conversations are imported into a topic map, the merge agent will ask if the two conversations are *about* the same subject. In this example, the subject is a question related to a causal factor in climate change. We see two subjects entailed in that question: *climate change*, and *carbon dioxide/co2*. Determination of subject sameness in this example relies on a synonym test.

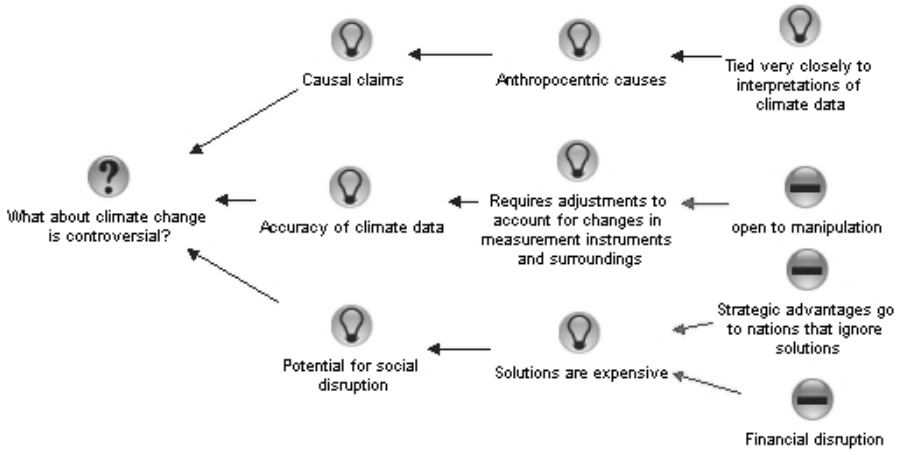



**Fig. 1.** Structured Conversation

[8] Compendium: http://compendium.open.ac.uk/

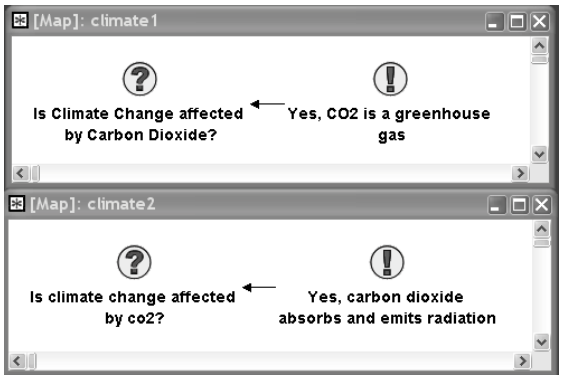



**Fig. 2.** Two IBIS Conversations

Our definition of conversation sameness centers on the context of the conversation. In the example, the context resides in a question, and that question, in both cases, is the same. Figure 3 illustrates a merged conversation.

For conversations of greater depth than illustrated, each node within the merged result is then treated as a merge subject and compared to its *siblings*. That is, each answer to each question is compared against other answers to the same question.

While synonym detection is a trivial lookup in a topic map, conversation merging remains a complex process. As an example, sentence structure is illustrated, again, by two sentences that ask the same question:

- How does carbon dioxide affect climate?
- How is climate affected by carbon dioxide?

Greater complexity entails technologies such as natural language processing (NLP) and emerging forms of *machine reading* (Etzioni, et al, 2007). Our research seeks to fabricate a platform with which to explore the territory of machine reading to support merge decisions in conversation federation.

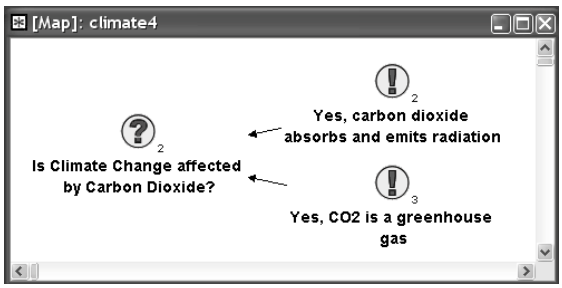



**Fig. 3.** A Merged IBIS Conversation

Merges, once performed, can be contested. Merge decisions must be transparent – merge rules that suggest a merge must inject their reasons into the merge process for later inspection. At the same time, merges should be capable of being reversed. We turn now to a look at merge technologies, one representative of traditional topic maps processes, and a new platform that answers questions similar to ours.

## 3 Related Work

In this section, we examine an instance of *algorithmic* merge detection and execution. We then examine a project that recognizes an important aspect of our desiderata, and implements a solution that is remarkably similar to the work we describe here.

Consider the merge determination algorithm from Ontopia's open source topic map platform[9], which we have chosen to examine since the platform is at once a popular and reliable instance of a commercial topic map product. In the following analysis, we examine two objects in Ontopia's XML topic map platform; the two objects examined are considered the equivalent of property objects defined by the TMRM document (Durusau et al., 2007). The Ontopia merge algorithm tests for overlaps between *subject identifiers* of two topics being compared, or overlaps between *item identifiers* of those two topics, or overlaps between *subject identifiers* and *item identifiers* of the two topics. Failing those tests, a further test asks if both topics *reify* the same object. Specifically, these tests are found in the Ontopia source code in this class:

```
net.ontopia.topicmaps.utils.MergeUtils
```

and implemented in this method in that class:

```
public static boolean shouldMerge(TopicIF t1, TopicIF t2)
```

The method returns the value true if topic t1 is determined to be *about* the same subject as topic t2. In more detail, the algorithm relies on two particular property types: *item identifiers*, and *subject identifiers*. We shall skip the reification process since it is not represented in TMRM implementations with which we are familiar. From (Garshol & Moore, 2008), a subject identifier is indirectly defined:

> "information resource that is referred to from a topic map in an attempt to unambiguously identify the subject represented by a topic to a human being"

A subject identifier bears strong resemblance to the Internet's Uniform Resource Identifier (URI) (Berners-Lee et al., 1998). An example of a subject identifier for a particular term defined in the TMDM is this:

```
http://psi.topicmaps.org/iso13250/glossary/association-role
```

The TMDM similarly defines an item identifier:

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

[9] Ontopia: http://code.google.com/p/ontopia/

For completeness, we borrow two more definitions from the TMDM:

- information resource: "a representation of a resource as a sequence of bytes; it could thus potentially be retrieved over a network"
- locator: "string conforming to some locator notation that references one or more information resources"

An information resource can be a URI as illustrated, or a URL – a direct address on the Internet of some web page of interest. Defined as string objects, an information resource is open to simple comparisons by the algorithm. Simple string comparisons are primary engines of subject identity comparison in the Ontopia algorithm.

When the Ontopia platform decides to merge two topics, one is chosen as a recipient, the other as a donor. There is then the equivalent of a *set union* of resources performed, where resources taken from the donor are included, without duplication, into the recipient. The donor is then removed from the map, leaving one topic that represents the sum of all resources known to that map as representations of that subject.

Aki Kivelä (2010) mirrors an aspect of topic merging reflected in our desiderata when he says:

> "After merge it is however impossible to solve where some piece of information originated from."

The Kivelä comment becomes important when precise provenance is required to create a view that separates, say, sources of information. Further, it is sometimes the case that a topic merge is later contested; the merge might have been triggered by a misinterpretation of or by false information. When such situations exist, a case is made to find an alternate solution to merge platform architectures. The Wandora[10] platform posits a layered topic map architecture (Kivelä, 2010) that is remarkably similar to the *virtual proxy* solution we describe below. Figure 4 illustrated an added proxy that contains the *subject identifiers* from two proxies that have been merged.

Figure 4, as we shall see, is remarkably similar to the virtual merge approach we take, as described next.

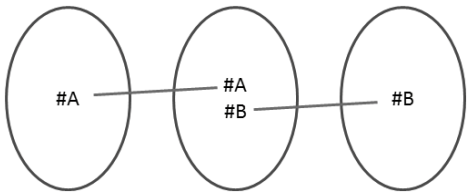



**Fig. 4.** Layered Merged Topics – After (Kivelä, 2010)

[10] Wandora: http://www.wandora.org/wandora/wiki/index.php?title=Main_Page

## 4 A Research Solution

We begin this inquiry by asking this question: *Why perform set-union merges*? We acknowledge one well-known reason: the result is a single object which consumes less space in the map and which provides the equivalent of a *joined query result* when fetched from a database. That is, the merge operation co-located all resources necessary to reconstruct the subject when requested. To visualize the alternative – that situation where no set-union was performed, meaning there are multiple objects representing the same subject in the database – a *join* operation must be performed either inside the database or following multiple queries to gather all related resources. That is, indeed, an important consideration.

We borrow and adapt a concept from the Ted Nelson *Xanadu* play book, his *virtual file* architecture (Nelson, 1999). The virtual file concept entails a large body of text created in a persistent way, and a *file* that is created as a document that contains a list of *links* into the large body of text (Fig. 5). If some text is to be modified, new text is created at the end of the large body, and appropriate pointers in the virtual file are adjusted.

In the simplest expression of Nelson's virtual file concept, we implement a *virtual proxy* that serves as a *binding point* for all merged proxies (Fig. 6) for a given subject. In this implementation, one creates *merge assertions* (associations) that specify the nature of the merge. For instance, one proxy is designated the *original proxy* assertion and another is designated a *merged proxy* assertion; in each case, the *justifications* for the merge are presented in the merge assertions – rounded rectangles that connect each subject proxy to the virtual proxy. Since each merge assertion is, itself, a subject, each merge is thus a candidate for social intervention in a contested domain; each merge assertion is *contestable*. The virtual proxy itself gains a set union of *subject identity* properties from each merged proxy. The virtual proxy becomes the core target for queries that seek subjects. When timestamps are included in the merge assertions, one gains a *temporal* view of the history of a map-mediated subject.

A virtual proxy allows us to maintain the separate identities of different representations of the same subject. In the case of merging representations of users that come into a federation with different object identities, we allow the separate identities to

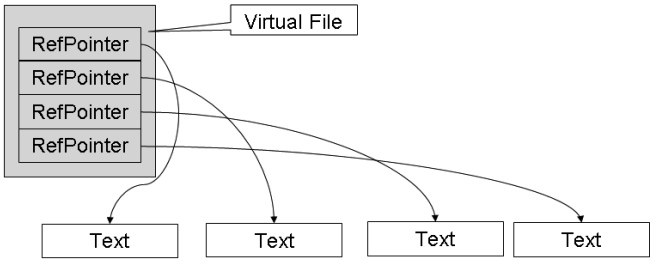



**Fig. 5.** Virtual File System after (Nelson, 1999)

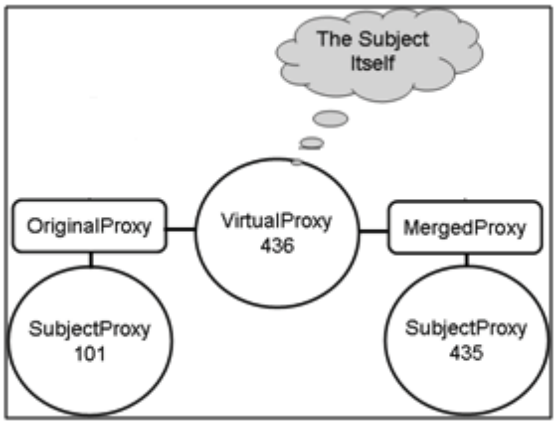



**Fig. 6.** Virtual Merge Graph

remain to facilitate continued actions of algorithmic processes. Views requested on those participants are joined together as needed.

In the scenario where contested merges call for reversing a particular merge, the merge assertion can be unlinked to release the contested proxy from its merge. Depending upon federation requirements, it may be reasonable to re-link that merge assertion into a subject that federates contested merges for purposes of historical analysis.

Performance aspects of information retrieval are in play when one plans for a federation infrastructure that entails merging resources. Specific to our interest is the cost of *joins* associated with production of views. In the set-union merge as described for the Ontopia algorithm, the join occurs during the merge process – the join is performed one time only. In the case of a virtual merge algorithm, joins are expected to occur during view production, on demand, with potential impact on system performance. At the implementation level, it is reasonable to consider performing a join just once and sequestering it in the virtual proxy. If a proxy merge is reversed, the join will need to be re-performed at that time.

## 5 Concluding Remarks

We believe that our virtual merge architecture enables the federation of complex topics, those associated with human conversation. The enabling factors include maintenance of provenance and the ability to reverse a merge already performed. By coupling subject proxies with merge assertions that are, themselves, subjects, we promote full transparency of merge operations and facilitate debates related to each merge.

The ability to request views *by author*, that is, to view a conversation while excluding certain participants, provides greater flexibility in user control of views. A reputation and trust system added to a federation can provide metrics by which individuals can be excluded from views by setting thresholds in a view request.

The Bloomer platform is now at the full prototype stage, available for download as an open source project. We are beginning installations in several international settings that will provide us with early experience and feedback with which to improve the system and validate our virtual merge architecture.

## References


1. Berners-Lee, T., R. Fielding, and L. Masinter (1998). Uniform Resource Identifiers (URI): Generic Syntax. Network Working Group: Request for Comments 2396. Online at http://www.ietf.org/rfc/rfc2396.txt
2. Bowker, Geoffrey, and Susan Leigh Star. (1999). Sorting things out: classification and its consequences. Cambridge, MA: MIT Press
3. Conklin, Jeff, Albert Selvin, Simon Buckingham Shum, and Maarten Sierhuis (2003). Facilitated Hypertext for Collective Sensemaking:15 Years on from gIBIS. Keynote Address: Proceedings LAP'03: 8th International Working Conference on the Language-Action Perspective on Communication Modelling, H. Weigand, G. Goldkuhl and A. de Moor (Eds.) Tilburg, The Netherlands 1–2 July 2003
4. Conklin, Jeff (2005). Dialogue Mapping: Building Shared Understanding of Wicked Problems. Wiley
5. Durusau, Patrick, Steve Newcomb, and Robert Barta (Editors) (2007). Topic Maps Reference Model, 13250-5. Online at http://www.isotopicmaps.org/TMRM/TMRM-7.0/tmrm7.pdf
6. Etzioni, Oren, Michele Banko, and Michael J. Cafarella (2007). Machine Reading. Proceedings of the 2007 AAAI Spring Symposium on Machine Reading
7. Garshol, Lars Marius, and Graham Moore (Editors) (2008). Topic Maps—Data Model. Online at http://www.isotopicmaps.org/sam/sam-model/
8. Kivelä, Aki (2010). Introduction to Layered Topic Maps. Online documentation for the open source Wandora topic map platform. Online at http://www.wandora.org/wandora/wiki/index.php?title=Introduction_to_Layered_Topic_Maps
9. Nelson, Theodor Holm, (1999). Xanalogocal structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use. ACM Computing Surveys, Volume 31, Issue 4es, December, 1999
10. Park, Jack (2008). Knowledge Gardening as Knowledge Federation. In Proceedings Knowledge Federation 2008: First International Workshop on Knowledge Federation, Dubrovnik, Croatia. October 20–22, 2008
11. Park, Jack (2010). Boundary Infrastructures for IBIS Federation: Design Rationale, Implementation, and Evaluation. PhD Thesis Proposal kmi-10-01. Knowledge Media Institute, The Open University, Milton Keynes, UK. Online at http://kmi.open.ac.uk/publications/techreport/kmi-10-01
12. Pepper, Steve, and Graham Moore (Editors) (2001). XML Topic Maps (XTM) 1.0. Online at http://www.topicmaps.org/xtm/
13. Rittel, H., and M. Webber (1973) Dilemmas in a General Theory of Planning. Policy Sciences, Vol. 4, 155–169, Elsevier Scientific Publishing Company, Inc., Amsterdam
14. Star, Susan Leigh. (1989). The Structure of Ill-Structured Solutions: Heterogeneous Problem-Solving, Boundary Objects and Distributed Artificial Intelligence. In M. Huhns and L. Gasser, eds. Distributed Artificial Intelligence 2. San Mateo, CA: Morgan Kaufmann Publishers. 37–54

# Et Tu, Brute? Topic Maps and Discourse Semantics

Lars Johnsen

University of Southern Denmark
Engstien 1, 6000 Kolding
Denmark
larsjo@sitkom.sdu.dk

**Abstract.** Arguably, to count as a truly semantic technology Topic Maps should not only be capable of encoding the meaning of individual propositions but also the rich semantics of discourse, i.e. the kind of information flows that we associate with running text or fluent speech. In this article, it is discussed how various forms of discourse meaning may be represented in topic maps. More specifically it is suggested how propositional, grammatical, modal and textual meaning may be encoded using a layered approach. In addition, it is briefly mentioned where "discourse topic maps" may be put to use.

## 1 Introduction

Topic Maps is sometimes described as a *semantic* technology or model. One reason is presumably that it represents information and knowledge in forms that are (thought to be) akin to the way humans structure meaning in sentences. A sentence such as "Brutus killed Cesar with a dagger" is naturally translated into a topic map structure like:

```
kill(agent : brutus, goal : cesar, instrument : dagger)
```

However, needless to say, much information, and most information meant for human consumption, does not come in simple self-contained statements, or propositions to use a slightly more technical term, like this one but in more or less complex and semantically intricate text-like forms ranging from entire books to extremely brief and compact messages like postings on micro-blogging services like Twitter. Consider for example the following bit of fictitious text expanding slightly on the Brutus-killed-Cesar-with-a-dagger theme:

> *Clearly, although Cesar must have seen, when entering the Senate in Rome on the Ides of March in 44 BC that Brutus, aka Marcus Junius Brutus, was cheesed off because he had chatted up Porcia, Brutus' wife, he did not expect that he might be brutally murdered by Brutus with a dagger. He had, therefore, not bought Band-Aid at Forum Romanum, although the market was open on that day, it being a weekday.*

Although this text in fact only contains two main sentences, it manages to weave together, as it were, a host of meanings and "semantic threads" into what most of us would probably regard as a pretty cohesive and coherent information flow, or piece of discourse. For instance:

- Despite its brevity, the text succeeds in conveying quite a few individual propositions (Cesar entered the Senate, Brutus was angry, Porcia was married to Brutus, etc.) and it is done through diverse grammatical means such as finite clauses, non-finite clauses and genitive constructions.
- Some of these propositions are neatly embedded in each other and play specific thematic roles in the action unfolding. They may be identified as "time" (when . . . ), "reason" (because. . . ) and "concession" (although . . . ).
- The two main parts of the discourse are linked in a kind of semantic relation, which may best be described as a "cause – effect" or "reason – result" relation: *Because* he did obviously not expect a murderous attack, Cesar had not bought Band-Aid at Forum Romanum.
- Actions and states of affairs are not presented as mere facts but are qualified in various ways. Cesar's murder is suggested as a possibility ("might") and his perception of Brutus' anger as a necessity ("must"). And the adverb "clearly", placed prominently at the beginning of the sentence, indicates the speaker's degree of belief in what he is uttering.
- Actions and states of affairs are seen from one perspective only, and that's Cesar's, even in the clause where he is the object or goal of the action. Here the passive voice is employed to retain the intended perspective ("he might be brutally murdered").
- Different readings of the text may yield subtle differences in meaning. The dagger, for instance, may either be seen as an instrument used in the killing of Cesar or, less obviously perhaps, as an object or property associated with Brutus (try and replace "with a dagger" with "with a broad smile").
- Flow is, to a high degree, achieved through direct links in the text. "It" refers back to "that day" which in turn refers back to "the Ides of March in 44 BC". And "the market" is a direct continuation of "Forum Romanum" denoting the same entity.
- The text has a certain informal ring to it especially because of expressions like "cheesed off" and "chatted up".

## 2 Meaning and topic maps

Now, one may argue that to count as a truly semantic (web) technology or model, Topic Maps must be able to capture and represent interacting meanings like these. In other words, Topic Maps should not only be capable of encoding the meaning of individual propositions but also the rich semantics of *discourse*, i.e. the kind of information flows that we associate with running text or fluent speech. This in turn requires the capability to simultaneously represent information about some possible world or domain, about a speaker's or writer's view of that world and last but not least about the way the speaker or writer has chosen to convey this information on a particular occasion.

Interestingly, the term discourse is not alien to the Topic Maps paradigm. In the Topic Maps Data Model subjects are described as "anything about which the creator of a topic map chooses to *discourse*" (Garshol & More, 2008) and topics are defined as representing "subjects of *discourse*" (ibid.) (my italics).

Below it is suggested how various types of interacting meaning in discourse may be modelled in topic maps. The discussion is loosely based on insights and approaches

in current frameworks within the fields of linguistics and discourse theory, notably Functional Grammar (Dik, 1997) and Renkema's Connectivity Model (Renkema, 2009). The approach proposed is based on two premises:

1. Modelling should, to the extent this is appropriate and useful for the purpose at hand, reflect the layered nature of natural language and discourse.
2. The various types of discourse meaning should be modelled in a consistent fashion based on explicit principles or criteria.

To start off, it may be useful to categorize discourse meanings into four broad types:

- Propositional meaning
- Grammatical meaning
- Modal meaning
- Textual meaning

*Propositional meaning* comprises the aspects of meaning that relate to situations, events and state of affairs and the entities taking part in these (e.g. Brutus, Cesar, kill, dagger). Propositional meaning is based on open classes which may be continually expanded (i.e. nouns, verbs, adjectives).

*Grammatical meaning* is meaning attached to propositional meaning through grammatical function words, inflectional endings and so on (e.g. kill*ed*, *was* killing, *will* kill). Grammatical meaning is realized by members of closed classes which cannot be extended (e.g. present/past/future, singular/plural, definite/indefinite, etc.).

*Modal meaning*, or speaker meaning, is semantics added to propositions to indicate the speaker's or writer's belief in, or attitude towards, what he or she is communicating (Brutus *may* have killed Cesar; *undoubtedly*, Brutus killed Cesar). Modal meaning may either be manifested through closed classes such as modal verbs (must, may, ought to, etc.) or through open classes like adverbs (e.g. undoubtedly, surely, probably)

*Textual meaning* is meaning that makes discourse fragments "hang together", so to speak. Textual meaning is often realized through cohesion, textual links such as connectors (because, although, etc.) connecting adverbs (therefore, however, etc.) or the use of pronouns (Brutus ... he, the dagger ... it).

In the model outlined here, the following guidelines are proposed:

- Propositional meaning typically involves *real world entities, events and situations* and should therefore, as default, be encoded using topics and associations.
- Grammatical meaning typically represents *properties* of real world entities, events and situations and should therefore, as default, be encoded using internal occurrences.
- Modal meaning typically reflects the *validity of statements* on the part of the speaker or writer and should therefore, as default, be encoded using scope.
- Textual meaning is typically used to facilitate the *elaboration, enhancement or extension of information* and should therefore, as default, be encoded using reification.

## 3 Propositional meaning in topic maps

In topic maps, propositional meaning is often encoded as associations and topics playing certain semantic, or thematic, roles. Consider the proposition "Brutus killed Cesar mercilessly with a dagger in the Senate on the Ides of March". It might look something like this:

```
kill(agent : brutus, goal : cesar, instrument : dagger,
     manner : mercilessly, place : senate, time : ides)
```

This representation, however, does not fully reflect the inherent sense of "kill" and its associated roles which are not necessarily on the same semantic level. Thus, one may argue that the killer and the victim are more central to the act of killing than the rest. Furthermore, it could be said that the manner of the act and the instrument used are also more closely linked to the act itself than the time and place. This point is best illustrated by the fact that in normal discourse the roles cannot naturally occur in a random order: We may have:

*Brutus killed Cesar mercilessly with a dagger in the Senate on the Ides of March*

and

*Brutus killed Cesar mercilessly with a dagger on the Ides of March in the Senate*

But not:

** Brutus killed Cesar in the Senate on the Ides of March with a dagger mercilessly.*

We shall therefore propose an approach allowing propositional meaning to be encoded in a layered structure. For instance:

```
kill(agent : brutus, goal : cesar) ~brutus-kill-cesar

#Brutus kill Cesar

proposition(nucleus : brutus-kill-cesar, manner : mercilessly,
            instrument : dagger)
           ~brutus-kill-cesar-mercilessly-with-a-dagger

#Brutus kill Cesar mercilessly with a~dagger

proposition(nucleus : brutus-kill-cesar-mercilessly-with-a-dagger,
            time : ides, place: senate)

#Brutus kill Cesar mercilessly with a~dagger on the Ides of
March in the Senate
```

(How appropriate the names given to roles and associations are in these examples may of course be debated but that's another matter).

These examples may be said to exemplify three forms of associations:

1. *Nuclear association* (association only with its "inherent" or "obligatory" roles, normally not more than three) – exemplified in a)
2. *Qualified association* (nuclear association + additional roles indicating semantic aspects of the act, event or state of affair: manner, instrument, direction, etc.) – exemplified in b)

3. *Located association* (qualified association + additional roles placing the act, event or state of affair in time and place) – exemplified in c)

Modelling layered semantics in this way in topic maps would have certain advantages. One would be the possibility to scope parts of a proposition. In a historical topic map on ancient Rome, for example, it would be possible to state as a fact that Cesar was killed by Brutus with a dagger but leave open the question if the murder actually took place on March 15 or indeed in the Senate.

## 4 Grammatical meaning in topic maps

Grammatical meaning couched in closed categories such as *tense* (kill vs. killed), *aspect* (killed vs. was killing), and *definiteness* (a dagger vs. the dagger) may be seen as properties of entities, events and situations and may therefore be modelled as internal occurrences. So "Brutus was killing Cesar with a dagger" might be represented in this way (some topics left out for the sake of clarity):

```
proposition(nucleus : brutus-kill-cesar,
            instrument : dagger ~instrument-dagger)
           ~brutus-kill-cesar-with-a-dagger

brutus-kill-cesar-with-a-dagger

tense: "past";

aspect: "imperfective".

instrument-dagger

definiteness: "indefinite".

dagger.
```

Here the occurrences "past" and "imperfective" are instances of the occurrence types "tense" and "aspect" respectively and attached to the topic reifying the association denoting the killing.

Likewise, "indefinite" is an instance of the occurrence type "definiteness" and attached to the topic reifying the dagger playing the role of instrument. This is in contrast to the generic concept of dagger which is unspecified with respect to definiteness.

An advantage of modelling grammatical categories using occurrences, rather than, say scope, is that it allows us to attach properties to reified role playing topics (such as "a dagger") and not just associations (such as "Brutus was killing Cesar").

## 5 Modal meaning

Modal meaning, or modality, is, as already noted, meaning indicating a speaker's belief in, or attitude towards, what he is uttering. Since modality may be said to indicate a kind of "valid context" for propositions, modal meanings are probably best represented as scope. For example:

```
kill(agent : brutus, goal : cesar) ~brutus-kill-cesar
@probability

brutus-kill-cesar

tense: "past ".

#Brutus may have killed Cesar

kill(agent : brutus, goal : cesar) ~brutus-kill-cesar
@likelihood

brutus-kill-cesar

tense: "future".

#Brutus will probably kill Cesar
```

Modality may be added to scope nuclear, qualified or located associations. However, it may also be attached to already modified statements. Take an utterance such as:

*No way, Brutus may have killed Cesar*

Here the speaker does not dispute the killing of Cesar by Brutus but the possibility of Brutus doing so. In other words, the speaker is modifying an already modified statement, or in Topic Maps parlance – scoping a scoped reified association. (This brings up the question if scope is, or should be, included in reifications or not).

## 6 Textual meaning

Textual meaning primarily emerges from propositions being linked in various ways. For instance:

*Brutus was cheesed off because Cesar had chatted up Porcia.*
*Cesar expected an attack. Therefore, he had bought Band-Aid*

It is clear that in these examples the propositions do not just follow each other sequentially but play specific roles in the semantic structure. The proposition designating Cesar's involvement with Porcia is the *cause* for Brutus' annoyance and the statement about Cesar's expectation is the *reason* why he bought Band-aid.

There may obviously be several ways of modelling semantic relations between propositions in topic maps but we shall propose one which reflects the manner in which these structures are either manifested in *in*dependent sentences or *de*pendent clauses. The first example "Brutus was cheesed off because Cesar had chatted up Porcia" may be represented as:

```
chat-up(agent : cesar, goal : porcia) ~cesar-chatting-up-porcia

cheesedoff(subject : brutus) ~brutus-cheesed-off
```

```
proposition(nucleus : brutus-cheesed-off,
            reason : cesar-chatting-up-porcia)
```

Here the reified association about Cesar chatting up Porcia is *embedded* as a topic playing the role of reason in the association stating that Brutus was cheesed off.

In the example about Cesar's foresight in buying Band-aid, two reified associations are joined in a discourse association which may be called "reason-result" or something similar:

```
expect(experiencer : cesar, experience : attack)
      ~cesar-expecting-attack

buy(agent : cesar, goal : band-aid) ~cesar-buying-band-aid

reason-result(reason : cesar-expecting-attack,
              result : cesar-buying-band-aid)
```

The premise here is that discourse is itself structured or layered. Take a very small text like this one:

*Brutus killed Cesar. Then he fled Rome. It was one mistake after another.*

It is clear that in this example the three propositions are not on a par. The last one functions as a kind of *evaluation* of two first two, while the two first are linked by a relation which may be labelled *time sequence*. The structure may therefore be encoded like this:

```
time-sequence(before : brutus-killing-cesar,
              after : brutus-fleeing-rome)
~brutus-killing-cesar-and-then-fleeing-rome

evaluation(evaluated: brutus-killing-cesar-and-then-fleeing-rome,
           evaluation: one-mistake-after-another)
```

In a discourse analysis framework like Renkema's connectivity model (Renkema 2009), the layered nature of discourse is described and analyzed using an extensive taxonomy of discourse relations. In this framework discourse relations not only serve to semantically link discourse segments like clauses and sentences but also indicate their informational significance within a certain stretch of text or speech.

## 7 But wait, there is more, a lot more ...

So far, we have talked about discourse meaning in terms of fairly broad categories, namely propositional, grammatical, modal and textual meaning. It goes without saying that these areas cover a range of discourse phenomena that need to be looked at more closely. In the small text above, we find things like:

- Appositions and genitives (e.g. Cesar had chatted up Porcia, Brutus' wife)

– Cohesion (e.g. the fact that “the Ides of March in 44 BC”, “on that day” and “it ” refer to the same thing and that “Forum Romanum” is also “the market”)

Appositions, genitives and cohesive devices are highly textual in nature and very much contribute to the flow of information in discourse allowing speakers and writers to (re-) introduce topics, place them in context, elaborate on them, etc. It is beyond the scope of this article to discuss in detail how phenomena like these may be modelled in topic maps but it seems that much can be achieved with reification in general, and the reification of role playing topics in particular, a feature seemingly not often employed in traditional topic maps.

For instance, to model the meaning of “Cesar had chatted up Porcia, Brutus’ wife” two associations may be formed, one denoting Brutus’ marriage to Porcia and one designating Cesar’s pass at her. In the latter, the topic of Porcia herself is not playing the semantic role of object or goal but rather the topic of Porcia playing the role of female spouse in the former:

```
married(husband : brutus, wife : porcia ~wifeporcia)

chat-up(agent : cesar, goal : wifeporcia)
       ~cesar-chatting-up-porcia
```

This not only links the two propositions tightly together but it also creates a subtle, but important semantic difference: not only does Cesar chat up Porcia, he in fact chats up the wife of another man!

The encoding of the cohesion between “the Ides of March”, “on that day” and “it” may also be done using reified role playing topics. The actual date (the Ides of March) may be represented as a topic playing the role of “time” in the association denoting Cesar’s entering the Senate. This role membership is then reified and taken up in two associations: one in which it plays a role to indicate the time when the market was open and another in which it is given the property of “weekday”. Note the semantic implications here: since “on that day” is a reification of a role membership it does not only imply that it was on March 15 that Forum Romanum was open but also on the very day Cesar entered the Senate.

## 8 Towards discourse topic maps

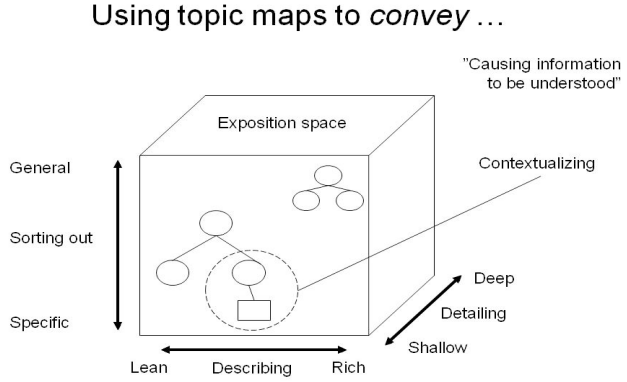
Now, why would anybody want to encode discourse structure in topic maps in the first place? Arguably, there are two major reasons:

*Discourse as topic maps (discourse is prior to topic map).* One is a wish or need to unambiguously represent or organize natural discourse occurring elsewhere, for instance in social media. An example of this would be a topic map charting one or more discussions going on among a group of people on Twitter. Here the representation of discourse meaning might have some kind of analytic, or even academic, purpose aimed at answering questions such as: how do these people interact, what clusters of persons or topics can be discerned, what information exchanges take place, what arguments are put forward, what linguistic strategies are put to use, etc. In fact, an area like semantic micro-blogging might greatly benefit from a systematic approach to discourse tagging,

as would perhaps other areas on what has come be known as the social semantic web (see Sayre 2009 or Johnsen 2010).

*Topic maps as discourse (topic map is prior to discourse)*. One reason for adding discourse structure to a topic map might be to enhance the communicative, or expository, value of its contents. For instance, in a topic map meant for e-learning one might link reified associations to indicate causal or temporal relations in its subject matter or to create "reading paths" for the user. In more general terms, discourse structure may be superimposed conceptual and content structures in topic maps to enhance intelligibility as well as findability on the web using a unified model.

In Johnsen (2010) a spatial metaphor is introduced to present topic maps as a means of codifying "discourse architecture". Discourse architecture may be seen as a hybrid between an organizational model such as a tree or a network classifying, describing and relating entities and their properties and running text in which situations are unfolded, contextualized and evaluated. These "expository" or "discourse" topic maps are construed as information spaces in three dimensions. The first dimension describes the act or result of *sorting out*, the activity of classifying subjects into types, subtypes and instances. This dimension may be said to go from general to specific. The second dimension represents the act or result of *describing*, assigning characteristics to topics. These descriptions may range from lean to rich. The third dimension constitutes the act or result of *detailing*, or *discoursing*, adding new layers of detail to already existing information structures in the topic map. Thus a description in a topic map may either be relatively shallow or deep depending on how much it has been elaborated upon. In addition to these three dimensions, the notion of contextualization is included to capture the act or result of scoping.



## References

1. Dik, S.C. (1997): The theory of Functional Grammar. Berlin: Mouton de Gruyter
2. Garshol, L.M (2004): Metadata? Thesauri? Taxonomies? Topic Maps! http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html
3. Garshol, L.M. & Moore, G. (2008) (eds.): Topic Maps – Data Model. ISO/IEC IS 13250-2:2006: Information Technology – Document Description and Processing Languages. International Organization for Standardization, Geneva, Switzerland. http://www.isotopicmaps.org/sam/sam-model/
4. Johnsen, L. (2003): Designing Adaptive Documentation with XML: From Formal to Rhetorical Markup. Best Practices, Volume 5, Number 3
5. Johnsen, L. (2010): Topic Maps: from information to discourse architecture. Journal of Information Architecture. Vol. 2, No 1
6. Park, J. & Hunting, S. (2003): XML Topic Maps. Creating and Using Topic Maps for the Web. Boston: Addison-Wesley
7. Pepper, S. (2010): Topic Maps. Encyclopedia of Library and Information Sciences. Third Edition, 1: 5247–5259
8. Renkema, J. (2009): The Texture of Discourse. Amsterdam/Philadelphia: John Benjamins Publishing Company
9. Sayre, J. (2010): A Flock of Twitters: Decentralized Semantic Microblogging. http://jeffsayre.com/2010/02/24/a-flock-of-twitters-decentralized-semantic-microblogging/

# Part VIII

# Topic Maps on the Web

# Extending Content Management with Topic Maps – Ontopia/Liferay Integration

Lars Marius Garshol[1] and Matthias Fischer[2]

[1] Bouvet ASA, Oslo, Norway
larsga@bouvet.no, http://www.bouvet.no
[2] HTW Berlin
matthias.fischer@fhtw-berlin.de, http://www.htw-berlin.de

**Abstract.** Building a Topic Maps portal should ideally be possible by simply installing some software, then clicking around in a user interface to set up the desired structure. This paper describes an integration of the Ontopia Topic Maps engine with the Liferay Portal and CMS to achieve precisely this.

## 1 Introduction

The main usage area for Topic Maps so far has been building web portals. So many Topic Maps-based portals have been built over the years that no complete list exists any more, but the number is probably in two digits, possibly three. Several Topic Maps tools exist for building such portals, but so far the Ontopia suite has lacked proper support for building portals. This paper describes an attempt to provide such a tool by integrating Ontopia with Liferay.

Liferay is a Java-based enterprise portal and content management system (CMS) based on the JSR-268 portlets specification. It is a commercial product of Liferay, Inc., but is also available under open source licenses.

Topic Maps are the ideal supporting technology for web portals, providing a conceptual and technical framework for navigation, search, classification, and interaction design. However, Topic Maps software generally does not provide content functionality, such as text formatting, image handling, versioning, multilinguality, workflow, and so on.

There is an infinity of content management software (CMS) on the market, which does provide all the described content functionality and more. However, CMSs generally have very poor metadata support. At best, there is support for a configurable set of text fields (possibly with some data typing), and perhaps a simple taxonomy module. Relations are generally not supported at all.

The purpose of integrating a Topic Maps engine such as Ontopia with a CMS is to provide the best of both worlds: the advanced metadata support in Topic Maps with the content features of a CMS, without having to write it all from scratch. Further, through the integration with the portal framework, we also make it easier to build Topic Maps-based portals using Ontopia.

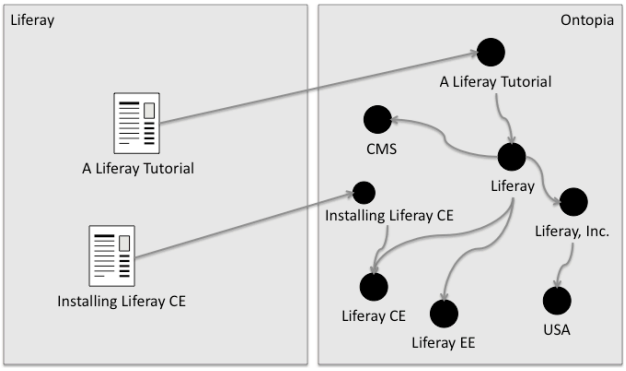



**Fig. 1.** Integration architecture

## 2 The CMS Integration

The purpose of the integration with Liferay's CMS is to allow the content in the CMS to be described in the topic map. For this to be possible, the content in the CMS must be represented as topics in the topic map. To automate this, and automatically maintain basic information in the topic map, a data integration has been developed. The integration uses the Liferay event API to update the topic map each time content is modified in the Liferay CMS.

Figure 1 shows an example, with Liferay web content objects on the left, and the corresponding topic map on the right. Only two of the topics represent Liferay content objects; the rest are used to describe the content.

### 2.1 Data integration

The data integration is a hook into the Liferay event API which intercepts Liferay events to maintain a Topic Maps representation of the content in Liferay. The representation is fairly straightforward: users are represented by user topics, communities by community topics, web content by web content topics, and so on.

The integration uses a semi-static Liferay Topic Maps ontology, which it relies on being present in the topic map. The ontology looks as follows in TMCL (using CTM syntax):

```
lr:user isa tmcl:topic-type;
  has-subject-identifier(1, 1, ".*");
  has-name(tmdm:topic-name, 1, 1);
  plays-role(lr:creator, lr:created_by, 0, *).

lr:wikinode isa tmcl:topic-type;
  # represents an entire wiki
  has-subject-identifier(1, 1, ".*");
  has-name(tmdm:topic-name, 1, 1);
  has-occurrence(lr:create_date, 1, 1);
```

```
  has-occurrence(lr:modify_date, 0, 1);
  has-occurrence(lr:lastpostdate, 0, 1);
  has-occurrence(lr:wikinodeid, 1, 1);
  plays-role(lr:creation, lr:created_by, 1, 1).

lr:wikipage isa tmcl:topic-type;
  has-subject-identifier(1, 1, ".*");
  has-name(tmdm:topic-name, 1, 1);
  has-occurrence(lr:create_date, 1, 1);
  has-occurrence(lr:modify_date, 0, 1);
  has-occurrence(lr:wikinpageid, 1, 1);
  plays-role(lr:creation, lr:created_by, 1, 1);

  # reference to containing wikinode
  plays-role(lr:containee, lr:contains, 0, *);

  # parent/child relation for wiki pages
  plays-role(lr:child, lr:parent-child, 0, *);
  plays-role(lr:parent, lr:parent-child, 0, *).

lr:webcontent isa tmcl:topic-type;
  # common superclass for all content types
  is-abstract();
  has-subject-identifier(1, 1, ".*");
  has-name(tmdm:topic-name, 1, 1);
  has-occurrence(lr:create_date, 1, 1);
  has-occurrence(lr:modify_date, 0, 1);
  has-occurrence(lr:review_date, 0, 1);
  has-occurrence(lr:expiry_date, 0, 1);
  has-occurrence(lr:display_date, 0, 1);
  plays-role(lr:work, lr:created-by);
  plays-role(lr:work, lr:has_workflow_state).

ir:workflow_state isa tmcl:topic-type;
  has-subject-identifier(1, 1, ".*");
  has-name(tmdm:topic-name, 1, 1).
```

In other words: Liferay users, wiki nodes (that is, wikis), wiki pages, and web content are all mirrored in the topic map. Every time you create, modify, or delete one of these the integration receives an event and updates the topic map accordingly.

In Liferay one can configure different so-called “structures” for web content. Each structure has a set of fields which are configured by administrators. For example, a typical news article structure might have the fields “title”, “abstract”, and “content”, where the abstract is a short summary shown on the front page and in search listings, while the content field holds the full article.

For each structure in Liferay the mapping automatically creates a new subtype of `lr:webcontent`, which is why the topic type is marked as being abstract. Individual web content objects are typed after which structure they use.

This approach makes it possible to handle content belonging to different structures differently in the topic map.

### 2.2 An example configuration

Let's say that we have a single structure, called `article`, which has the three fields we described above. Once we configure it, the topic map will contain the following (in CTM):

```
article ako lr:webcontent;
  - "Article".
```

To be able to describe articles in the topic map we need a bit more. Let's say that we make an `is-about` association type, which can connect to topics of type `category`. In practice this is probably much too simple, but for an example it will do. We then create the following in the Ontopoly Topic Maps editor that is part of Ontopia:

```
article
  plays-role(lr:work, is-about, 0, *).

category isa tmcl:topic-type;
  has-name(tmdm:topic-name, 1, 1);
  plays-role(subject, is-about, 0, *).
```

A graphical illustration of the Liferay ontology extended with a site-specific ontology can be seen in Fig. 2.

### 2.3 User interface integration

The data integration is only the first step in a useful CMS integration, because the only thing it does is to mirror the CMS contents in the topic map. The real purpose here was

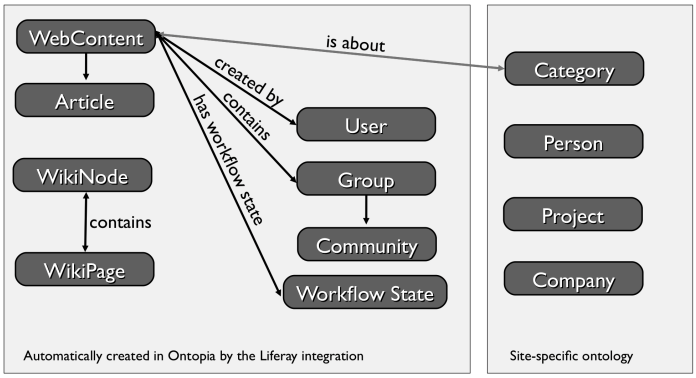



**Fig. 2.** Ontology

to allow users to describe their CMS content with Topic Maps, and for that we need to go beyond what the CMS can already do.

Of course, once a content object is created in the CMS it is possible to find the topic for it in the Topic Maps editor and add the `is-about` associations there. However, this is both awkward and difficult for users, who then have to deal with two different tools, and have to switch between them to perform what should be a single task.

The user interface integration solves this by allowing CMS users to describe their content in the topic map from within the same user interface that they use to author the content. This is done by integrating the Ontopoly Topic Maps editor in Ontopia into the user interface of the Liferay CMS.

So when users create a content object in Liferay the Ontopoly page for that content object's topic is displayed inside Liferay, as shown in Fig. 3. The fields shown for the content are those configured for topics of that type in the ontology.

Thus, topics of type `article` may be set up in Ontopoly to have fields like name (title from the CMS), author (set by CMS), date created (set by CMS), is about (set by author), and so on. When editing an article in Ontopoly all these fields will be shown. However, in Ontopoly it is possible to define a view, say "the Liferay CMS view", which

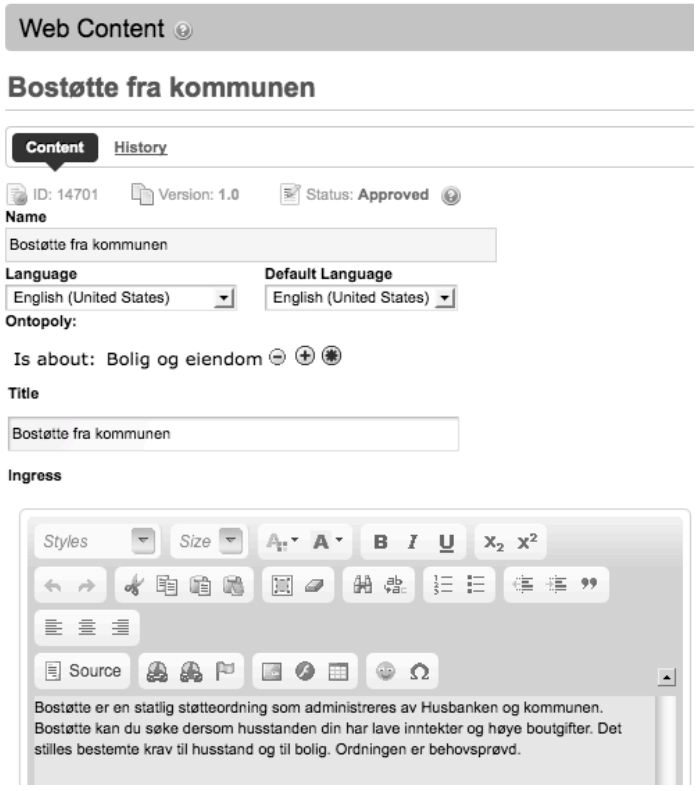



**Fig. 3.** Web content UI

contains only the "is about" field. The integration can be configured to use this view, so that in the Liferay CMS only the "is about" field will be shown in Ontopoly (as the other fields are already shown and set by the CMS and don't need to be duplicated).

Further, since different types of content map to different topic types in the topic map, it's possible to have different fields for different types of content. All of this is easily configurable using the ontology editor part of Ontopoly.

## 3 The Portal Integration

So far, all we have done is to make it possible to describe Liferay content with Topic Maps, and while that is of course valuable in itself, we also need to be able to build portals based on that Topic Maps description. The portal part of Liferay provides very powerful support for this, making it possible to create and populate pages simply by dragging and dropping pieces of functionality known as portlets into them.

In order to be able to display Topic Maps information in the portal we need portlets which can pull information out of the Liferay topic map. Our goal is to provide a set of portlets which provide much of the common display functionality sites are likely to need. All real projects will probably need to develop some Topic Maps portlets of their own for the more customized parts of their portals, but this is easy to do using standard Java Server Pages (JSP) and portlets.

Currently the integration consists of five portlets, but more are being planned. The following sections each describe a single portlet.

### 3.1 Related topics

A very common feature of Topic Maps-based portals is a display of a topic's associations, and this is what the portlet displays. The input is a single topic, and if the topic were this paper, the output might be something like:

```
is about
  Ontopia
  Liferay
  CMS integration
published in
  Proceedings of TMRA 2010
written by
  Lars Marius Garshol
  Matthias Fischer
```

The portlet can be configured to modify the display in several different ways to provide more flexibility. There are mechanisms to control which topic types and association types are shown (for example, one can leave off the "written by" association because it's shown elsewhere on the page), how the headings are formed, how headings and topics are sorted, and so on.

The full configuration options are too complex to cover in this paper, but the component underlying the portlet has been used in several production projects and is known to support a wide range of usage scenarios.

### 3.2 Article list

Another common feature of Topic Maps portals is wanting to display a collection of articles from the CMS based on criteria from the topic map. For example, one might want to show all articles on a given topic, all articles by a particular author, or all articles this month on a specific topic or any of its subtopics, or ...

The article list provides a simple way to perform this, based on a tolog query and a Liferay template. The tolog query produces the list of articles to display, which means that developers can configure this portlet to use just about any imaginable criteria. Allowing the Liferay template to be used to be configured provides detailed control over what each individual article in the list looks like.

### 3.3 Dynamic web content

This is a portlet that really should be part of Liferay, but isn't. What it does is simply to display a Liferay content object using the Liferay template given in the configuration. Which content object is shown depends on parameters passed in the page URL.

This makes it possible to create Liferay pages which display different articles depending on query parameters in the URL. It is of course trivial, but also necessary in order to build a fully dynamic site.

At some point Liferay will probably provide a portlet for this, at which point this portlet can be retired, but until then it is actually necessary.

### 3.4 Topic name

This is the simplest portlet of all: it takes a topic ID from the page URL and displays the name of that topic as a heading. This makes it easy to build topic pages in Liferay. The portlet cannot at the moment be configured any further.

### 3.5 Yahoo tree

This portlet is used to provide a user-friendly view of the top two levels of a hierarchy. It is called Yahoo tree because this is the view that Yahoo.com showed of its hierarchical classification about 15 years ago. For each top-level node in the hierarchy it displays that node as a heading, and then its children below it. The whole is arranged into a grid of a configurable number of columns, and the number of rows is determined by the number of top-level nodes.

The configuration consists of a tolog query to find the top-level nodes, another to find the children of a given node, and the number of columns to display.

This is sufficient to provide a good start page view of a hierarchical classification.

### 3.6 Common configuration

An issue that recurs in several of these portlets is that they need to be able to link to pages for the topics they display. Usually, portals consist of a number of different page

templates for topics of different types, and so it is not appropriate to configure the linking as part of the settings of individual portlets. It is really part of the global portal structure.

Instead, we provide a "URL template" occurrence on topic types, which allow users to state that all topics of type "person" have the URL `/person?topic=%topicid%`. This is picked up by all the portlets, and causes them to link to the person page when displaying person topics.

### 3.7 Planned portlets

A number of portlets are planned, but not yet developed. The most important of these are:

- A search form portlet and a search result portlet. These would allow Topic Maps searches to be performed and displayed.
- A similar topics portlet. This would be used for example on pages for web content and show other web content about similar topics.
- A tree view portlet, which would show a expandable and collapsible tree view of a hierarchy.

## 4 An example application

In order to make this rather abstract discussion a bit easier to follow we present an example application in this section, showing how all the different pieces fit together. The site we are developing has a hierarchical classification imported into the topic map, and web content written in Liferay. The classification used is LOS, a taxonomy developed by a Norwegian government agency for use in municpal government portals.

The front page is shown in Fig. 4:

In a real installation it would have front-page articles and much more, all of which is trivial to do within Liferay itself. Here we use the Yahoo tree portlet to display the top of the classification hierarchy in five columns.

If we click on one of the topics we get to the topic page (shown in Fig. 5, which has the URL `/temaside?topic=T7011`.

There are three portlets in this page. The heading is the topic name portlet, showing the name of the current topic. Immediately below is the article list topic, showing all articles (with title and abstract) about this topic. On the right is the related topics portlet, showing the relations of the current topic. Note that the "is about" relation is not displayed, because display of it has been turned off in the portlet preferences.

Clicking on one of the article headings takes you to the article page, which is shown in Fig. 6.

This page consists of just two portlets: the dynamic content portlet on the left, used to display the actual article, and the related topics portlet on the right, used to display the associations of the article.

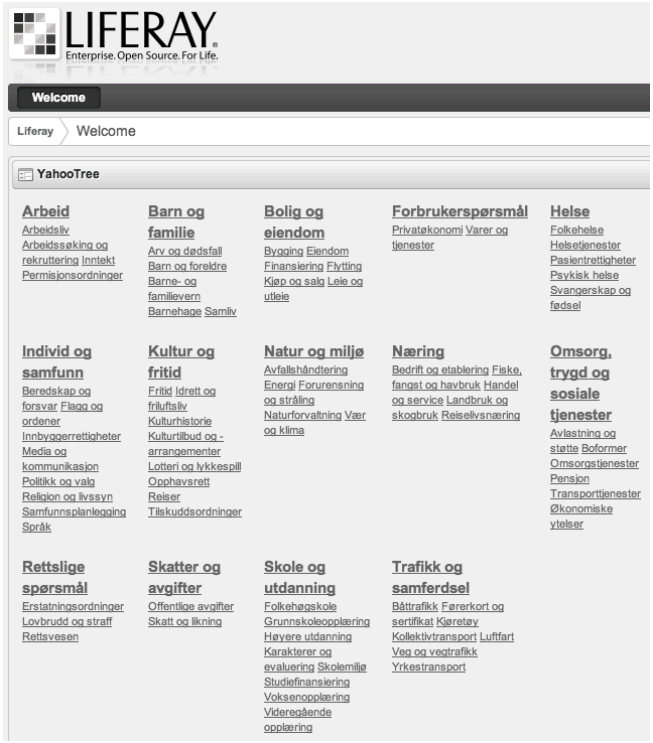



**Fig. 4.** Front page


**Fig. 5.** Topic page


**Fig. 6.** Article page

## 5 Conclusion

This paper has shown how a Topic Maps engine can be integrated into a CMS and portal in order to provide advanced metadata capabilities for a standard CMS, and at the same time greatly simplify site development with Topic Maps. Using the integration it really is possible to set up a portal by clicking around in Liferay and Ontopoly, and adding a couple of tolog queries in the configuration of some portlets.

Of course, in a real-life project more portlets will be needed than the rather limited set developed so far. This, however, is not a great obstacle to such a project. Some custom development is always to be expected.

# A PHP library for Ontopia-CMS Integration

Andrej Hazucha, Jakub Balhar, and Tomáš Kliegr

Department of Information and Knowledge Engineering
University of Economics, Prague
Nám. Winstona Churchilla 4, 130 67 Praha 3, Czech Republic
\{andrej.hazucha,xbalj20,tomas.kliegr\}@vse.cz

**Abstract.** This paper presents a library for integrating PHP-based content management systems with remote knowledge bases. Through the administration component the admin user defines queries, which are locally remembered and parameterized, and XSLT transformations that are used to visualize the results of the query against the knowledge base. The WYSIWYG editor-plugin allows the user to include the query into CMS documents. The library is currently integrated with Joomla! CMS system and was tested against the TMRAP interface of Ontopia Knowledge Suite, a SPARQL endpoint and a custom restful wrapper for Berkeley XML database. This work also presents an experimental domain-specific GUI-based query designer.

## 1 Introduction

Integration of Knowledge Bases (KB) with Content Management Systems (CMS) is an important practical as well as research topic. While CMS systems are used to create, organize and retrieve unstructured text, KB systems play the same role for structured content. The target application of the Knowledge Base Include (KBI) library presented here is to bridge the gap between these two types of systems by enabling access to the structured content stored in a KB from within a CMS. Specifically, the library allows the CMS user to query the KB; the query result is transformed into HTML and included into CMS documents where it can be easily combined with original CMS content.

The library presented in this paper was originally developed to enable the interoperability of the PHP-based Joomla! CMS system[1] and the Java-based Ontopia Knowledge Suite [3] (OKS) within the SEWEBAR project[2] to facilitate writing reports on data mining tasks [7]. However, the library is generic, easily portable to other CMS systems and interoperable with other types of knowledge bases.

The work presented here is organized as follows. Section 2 presents existing approaches to the CMS-KB integration. Section 3 introduces the KBI library. Section 4 briefs on the Joomla! implementation of the library. Section 5 describes how can the library be integrated with a domain-specific GUI query designer. Section 6 presents a short case study on the use of the library in the SEWEBAR project. Conclusions give account of drawbacks of the current implementation and highlight possibilities for improvement.

[1] joomla.org
[2] sewebar.vse.cz

## 2 Existing approaches to CMS – KB Integration

Due to heterogeneity of information stored in CMS systems, the organization of content into relational data structures is not flexible enough. This may be a reason why most of the research is exerted into bringing semantic web technologies to the CMS.

Interactive Knowledge Stack (IKS) [1] is a European project that aims to integrate semantic tools and CMS systems. The project cooperates with about 40 CMS vendors. Their joint objective is an architecture for easier integration of semantics and CMSs. The first implementation of this architecture is the JAVA framework FISE. A semantic editor that will allow user-friendly annotations and more features for semantic lifting is in development.

ITMS (Integrated Topic Management System) [5] is a concept for building CMS based on the Topic Maps standard where content would be considered as knowledge. ITMS is meant as extension to existing systems so that well known tools can be used. The main idea is to change hierarchically structured content nodes into Topic-Map like structures. The system would be connected with an ontology and content nodes are represent instances of topics from this ontology.

Similar concept is being implemented in popular CMS Drupal. Upcoming version of Drupal supports handling semantic information in RDF and RDFa [4].

The other approach is to enable semantics in wiki systems. The original MediaWiki[3] system which is used by wikipedia.org has *SemanticMediaWiki* extension which allows annotations, semantically enriched content management and structured querying in distributed knowledge. Another wiki system with semantic support is developed within the KiWi project [2]. This project creates a new engine with semantic tools based on RDF concept already in core. It means also that the point of view for content creation and manipulation is a little bit different.

While IKS is aimed at using semantic technologies for organization or annotation of documents in the CMS, in KBI library the content retrieved from the KB plays the same as role as the unstructured content. A similar notion is adopted by the KiWi system, which merges a semantic web RDF/OWL knowledge base and a CMS into one system. In contrast, in KBI the knowledge base is considered as a standalone system which runs on an arbitrary software architecture. No assumptions are made as of the availability of a specific communication protocol. The KBI library plays the role of information broker residing on the side of the CMS system, i.e. it does not duplicate any of the roles of the knowledge base.

## 3 The KBI Library

The Knowledge Base Include (KBI) library consists of communication procedures and CMS interfaces. This allows to use one implementation in various CMSs.

The core element of this library is the interface *IKBIntegrator* which defines the *query* method. This method is responsible for sending request query to the KB, receiving an answer and transforming its output if necessary. The inputs for the method are text of

[3] mediawiki.org

the query (generally just text, in case of OKS it is a tolog query) and optionally XSLT tranformation that is used for visualizing the results into an HTML fragment. In the library it is assumed that the KB returns the query result in XML.

The specific request method or the number of parameters passed may differ for various KB systems therefore the *query* method is KB dependent and should be implemented for every KB system used with the KBI library. There is also a generic implementation of this method inside the library using HTTP GET/POST and the SOAP protocol.

The KBI library can be customized if the generic SOAP and GET/POST query method are not supported by the selected KB system. This customization involves implementation of the query method and processing KB dependent parameters.

For example, OKS communicates remotely by the TMRAP protocol where queries are in the tolog language the and results have XML syntax. The KBI setting for TMRAP consists of: a tolog query, TMRAP interface URI (KB query source) and an XSLT from TMRAP result format to HTML. TMRAP is accessible via HTTP GET so it is possible to use the generic query method implementation.

## 4 Joomla! Integration

The KBI library contains code for CMS-KB interaction that can be shared by multiple CMSs to facilitate integration with a specific CMS. Most of the CMS specific parts is related to the user interaction. This section covers a description of KBI integration with Joomla! CMS, the world's most popular open-source CMS system.

Because of the way the extension in Joomla! CMS works specific KBI code was separated into several extensions. The core extension `com_kbi` takes care of definition of KB query sources, queries and related XSLT transformations. The KB source is actually instance of IKBIntegrator and provides UI for setting it up.

The editor-xtd plugin `kbInclude` extension allows to insert the HTML fragment representation of KB query results into a CMS document. The plug-in contains an UI window where the user selects the KB source, the query to run, sets its parameters and chooses the XSLT that will convert the query result to HTML. The resulting query specification can be pasted into the document (Joomla! article) and reexecuted every time the page is rendered (*dynamic insert*) or sent immediately via KBI Library to KB and only its result is inserted into the article (*static insert*). The format of query specification inserted into the article in case of dynamic query is a simplified JSON with plugin identifier.

The dynamically inserted query specification needs to be processed in a separate plugin (`kbi Content plugin`). This plugin executes the query during page rendering.

Figure 1 gives an overview of KBI-Joomla! integration.

## 5 Query Design Support

The Editor-xtd plugin `kbInclude` allows the user to execute one of the predefined queries. The goal of the user is to use the query to retrieve a specific kind of object from the database, whose properties and possibly relationships with other objects match

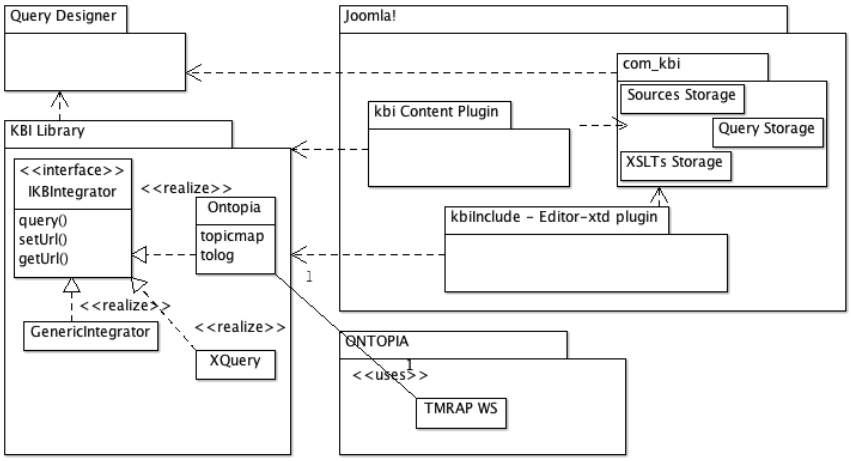



**Fig. 1.** System architecture

certain conditions. By default, the user selects the query from the list and then types in the values of its parameters. However, we are trying to make this workflow easy to use for domain experts. Therefore queries and their parameters can be built using Association Rule Query Designer. This visual designer from Fig. 2 is a PHP/JavaScript application built in our Joomla! extensions. The user constructs the query by dragging the building blocks preloaded from the knowledge base (A) into rule template (B).

The use of visual designer makes the internal workflow a little more complex. The query designer is initialized by configuration data and after user input it *generates* the query, which is processed and passed through the KBI library to the KB.

The configuration data of Query Designer with KBI involves: 1) a configuration XML that aligns the query designer expressiveness with the constraints imposed by the query generation XSLT (query dependent), 2) XML containing the codebook with names of data fields and data field values as appearing in the source Topic Map (topic map

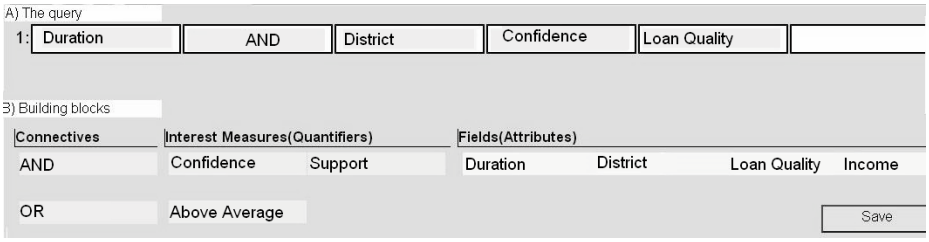



**Fig. 2.** The Association Rule Query Designer

dependent) and 3) an XSLT transformation that translates the XML produced by the

query designer to the query language provided by a KB. The configuration XML can be generated from within UI and contain data structures gained from remote KB.

## 6 Use Case

In this section, we present an on-going work on the Query Designer for the Association Rule Ontology topic map, which was introduced at last year's TMRA [6]. This topic map contains primarily association rules, which are patterns that were learned from some dataset using a machine learning algorithm.

Often, thousands of rules are learned, so the data analyst needs to hand-pick only the most useful rules to present to the bank. The final outcome of association rule mining in this task are not just these rules, but a report written in a CMS system which describes the task, existing domain knowledge and lists some of the discovered rules, which the analyst deems to be most valuable for the expert [7].

In SEWEBAR project, we use OKS to store the discovered association rules. The task for *KB Include* is then to allow the data analyst to issue a query from the CMS that will retrieve only a small subset of the large number of rules stored, for example rules involving good loan quality and low income.

The analyst can either issue the query directly from the editor plugin or with the query designer.

**Directly from editor** The KBI contains several predefined queries for association rules. The user chooses the query from the list and specifies its parameters. The problem is that the parameters typically pertain to names of topics – client properties – in the queried topic map, which the analyst will not generally remember in verbatim as required by the tolog query.

**Association Rule Query Designer** (ARQD) allows to formulate the query in a graphical manner. The ARQD is instantiated from the editor. The XML with the query built by the analyst in the ARQD is then forwarded to KBInclude, which transforms it with an XSLT transformation to a tolog query and issues this query via TMRAP against OKS. The result of the query is then transformed with XSLT stylesheet to an HTML fragment, which is included into the report along with the analyst's free text comments.

## 7 Conclusion

The KBI library introduced in this paper reacts to the increasing demand for integrating knowledge bases, particularly the semantic one, with ordinary CMS systems. From technological standpoint these two types of systems have often little in common: KBs are often Java based applications and the underlying hardware must be able to deal with complex queries. On the other hand, popular CMS systems are lightweight PHP application achieving fast response on commodity hardware.

The solution introduced in this paper to this dichotomy is built on the same lightweight philosophy as the typical open source CMS.

A possible bottleneck is the extensive use of XSLT transformations for query processing, which can be slower than PHP code. Compared to PHP code, XSLT brings higher security and due to declarative semantics better readability.

The most needed enhancement of the KBI library is the support for asynchronous querying as queries against semantic knowledge bases can be very computationally intensive. Since the query is included in design time into the CMS document, which is then accessed multiple times later, the natural solution is to issue the query off-line and cache the resulting HTML fragment.

The KBI library, its Joomla! implementation, the AR Designer (which is still in development), several sample tolog queries and the corresponding XSLT transformations are available at http://sewebar.vse.cz.

## Acknowledgment

This paper was prepared with the support of “Institutional funds for support of a long-term development of science and research at the Faculty of Informatics and Statistics of The University of Economics, Prague".

## References

1. Interactive knowledge stack. http://www.iks-project.eu
2. Knowledge in a wiki. www.kiwi-project.eu
3. Ontopia knowledge suite. http://ontopia.net
4. Stephane Corlosquet, Renaud Delbru, Tim Clark, Axel Polleres, and Stefan Decker.Produce and Consume Linked Data with Drupal. 2008
5. Lars Marius Garshol. Topic maps in content management. http://ontopia.net/topicmaps/materials/itms.html
6. Tomáš Kliegr, Ovečka Marek, and Zemánek Jan. Topic maps for association rule mining. In *Proceedings of TMRA 2009*. University of Leipzig, 2009
7. Tomáš Kliegr, Martin Ralbovský, Vojtěch Svátek, Milan Šimunek, Vojtěch Jirkovský, Jan Nemrava, and Jan Zemánek. Semantic analytical reports: A framework for post-processing data mining results. In *ISMIS'09: 18th International Symposium on Methodologies for Intelligent Systems*, pages 453–458. Springer, 2009

# JavaScript Topic Maps in Server Environments

Jan Schreiber

Ravn Webveveriet AS, Pb 2169 Grünerløkka, 0505 Oslo, Norway,
jans@ravn.no

**Abstract.** This paper shows how server-side Topic Maps applications can be created with *node* and the JavaScript Topic Maps engine *tmjs*. Node provides event-based JavaScript for server-side applications. As an example, a simple PSI server is presented. Node requires applications to run in one single thread using non-blocking, asynchronous function calls. To achieve this goal, this paper outlines an asynchronous version of TMAPI.

## 1 Introduction

Despite its bad image, JavaScript is an elegant, lightweight, and highly expressive language [7]. JavaScript includes a lot of ideas from functional languages like Lisp, but its syntax reminds more of the syntax of C. Virtually every personal computer in the world has at least one JavaScript interpreter installed on it and in active use [5]. This makes JavaScript an attractive platform for an Open-Source Topic Maps engine.

The idea of creating a Topic Maps engine in JavaScript is not new. In 2003, Thomas Passin and Alexander Johannesen released TM4Jscript [3]. However, since then Topic Maps has undergone a lot of changes, including the release of TMDM [13], and, in February 2010, TMAPI 2.0 [11], an object-oriented Topic Maps API for Java. TMAPI has been ported to other languages like PHP [8] and .NET [12]. This lead to the development of tmjs, a JavaScript Topic Maps engine that implements the TMDM with a TMAPI-like API.

The next section presents tmjs and its design principles. In the third section, a brief overview of node is given. The fourth section presents a simple PSI server as an example of a Topic Maps application based on node. The fifth section explains how asynchronous TMAPI can increase performance of such applications.

## 2 Overview of tmjs

tmjs is an Open-Source Topic Maps engine written in pure JavaScript, namely ECMAScript edition 3 (ECMA-262) [6]. The library is written in object-oriented style, and implements most of TMAPI 2.0. At the moment, the only back-end uses in-memory storage.

The implementation tries to be as portable as possible, and tmjs is mostly independent of its environment to increase the number of supported platforms. Particularly, tmjs does not use the Document Object Model (DOM), which is available in browsers, but not necessarily in server environments. It does not depend on external

libraries like jQuery or prototype.js either, but it can be used together with these libraries. Due to the use of namespaces only two globally available symbols, namely `TM` and `TopicMapSystemFactory`, are exported by the library. All other symbols are contained in the `TM` namespace.

The engine supports all modern desktop browsers, as well as browsers on mobile devices like the iPhone or Android phones, and JavaScript server run-time environments like Mozillas Rhino [9], Mozillas SpiderMonkey [10], and Googles V8 JavaScript Engine [14].

tmjs implements the full Topic Maps Data Model, including support for variants. It supports merging of topics. Auto-merging of duplicate associations and other topic map items is currently under development.

The provided API aims to be TMAPI 2.0 compatible, though minor changes to the original Java API had to be made. This is because JavaScript does not support method overloading. TMAPI makes excessive use of this feature, mainly in the `createOccurrence()` methods, but also in other places. Although this can be simulated by checking the types and number of arguments, it seems more natural for a JavaScript library to provide different function names.

Another notable feature is chaining of TMAPI calls. All functions that modify the topic map return the object they modify. This makes it possible to combine multiple function calls. `getParent()` can be used to retrieve the parent of the current topic map construct:

```
tm.createTopic().addSubjectIdentifier(foo).createName("bar").
  getParent().createOccurrence(type, "baz").
  addScopingTopic(quux);
```

To improve code quality, all unit tests of the TMAPI 2.0 project have been ported to JavaScript, and more unit tests have been added. The code validates with JSLint[1], a JavaScript equivalent of `lint` for C written by Douglas Crockford.

The following JavaScript code shows how to create a `TopicMap` object with `tmjs`:

```
var factory, sys, tmid, tm;
factory = TopicMapSystemFactory.newInstance();
factory.setProperty('com.semanticheadache.tmjs.backend','memory');
sys = factory.newTopicMapSystem();
tmid = sys.createLocator("http://example.org/mytm");
tm = sys.createTopicMap(tmid);
```

As indicated above, an in-memory back-end is implemented. Future releases will include a Web SQL database back-end, and possibly key/value back-ends for node.

tmjs provides a plug-in for JTM 1.0 import and export. An experimental plug-in for XTM 2.0 import and export is planned for the next release. The latter will be based on the browsers DOM implementation, and will not work on the server side.

[1] http://www.jslint.com/

## 3 JavaScript server applications with node

Server-side JavaScript has been available for over a decade. The first commercially available implementation was Netscape's LiveWire, released in 1996 as part of Netscape Enterprise Server 2.0.

Node (or node.js) was written by Ryan Dahl. It is a JavaScript wrapper around libevent for Googles V8 JavaScript engine. It can be seen as a framework for server applications, and it has built-in support for TCP, DNS, HTTP. Its primary goal is to provide an easy way to build scalable network programs. [2].

This is achieved by making all network I/O non-blocking and all file I/O asynchronous. That means that there is only a single thread of execution. All function calls that use I/O are therefore asynchronous. When such a function is called, it returns immediately, and the result of the function is returned asynchronously via a callback function.

This asynchronous model seems strange at first, but it simplifies programming opposed to multithreaded programming since the programmer does not have to deal with deadlocks or synchronization issues. It behaves exactly like JavaScript in a Web browser.

Node is an interesting framework for tmjs. Due to the extendability of tmjs, import and export plug-ins as well as storage back-ends can be created with library wrappers for node.

HTTP is a first class protocol in node, and HTTP servers serving information from Topic Maps can be created with only a few lines of code. Other areas for applications include Topic Maps based DNS servers, proxy servers or peer-to-peer applications. Node gives access to low-level POSIX services via JavaScript, which otherwise are limited to software written in C.

At the moment, a drawback of node is that the API is unstable and changes from release to release, which makes it necessary to often port application code to new releases of node.

## 4 Use case: A simple PSI server

As an example application, a simple server for published subject identifiers[2] has been created by the author of this paper. It is inspired by the PSI server at http://psi.ontopedia.net.

The idea is simple: The server application can be started with any topic map, and it will begin to serve information about all subject identifiers that belong to a given domain name. The server can be started from the command line with a given topic map file and a server name, and then starts serving pages for all subject identifiers that match the given domain name. The server reads the topic map on startup and keeps it in memory. Here is a simplified version of the main loop:

```
var tm, servername;
// assumes that the TopicMap object tm has been initialized and
// the topic map in question has been imported successfully.
```

[2] The full source code will be made available at http://github.com/jansc/node-psi-server

```
// servername is set from a~command line parameter
http.createServer(function (req, res) {
    var url = 'http://'+servername+req.url, loc, topic;
    loc = tm.createLocator(url);
    topic = tm.getTopicBySubjectIdentifier(loc);
    if (topic) {
        res.writeHead(200, {'Content-Type': 'text/html'});
        // create a~page with information about the PSI
    } else {
        // Not found: create a~404 response
    }
    res.close();
}).listen(80);
```

This is all that is needed to create a Web server that listens to port 80 and serves HTML pages with information about subject identifiers from the topic map `tm` that belong to the domain `servername`. This code snipped omits how the content of the HTML pages is created and what information they contain. The full PSI server has more features:

- More output formats based on HTTP Accept-headers: HTML, JTM, XTM
- Configurable list of occurrences and name types that are included on the generated pages (e.g. `dc:description`)
- Support of a black-list of PSIs that are being treated as non-existent although they belong to the domain `servername`
- Support for caching, logging, debug information via the Connect library[3]

## 5 Asynchronous TMAPI

As stated in the third section, node applications are single-threaded, and therefore all calls involving disk, network or another processes have to provide a callback function. In the current in-memory implementation of tmjs all API calls are non-blocking because no I/O is involved. However, this situation changes once file operations (e.g. reading a topic map from a file) or database queries (e.g. a database back-end) are implemented. In this case the present TMAPI implementation should be changed. This is best illustrated by a short example:

```
var topics = tm.getTopics();
```

The above function call either blocks the current process while the topics are being fetched from topic map or it implies multiple execution stacks. If a callback function is supplied, the function call can return to the event loop immediately, and the callback function gets called when the result of the original function call is available:

```
tm.getTopics(function (topics) { /* \dots */ });
```

In general, the function interfaces should be changed to

```
obj.method(arg1, arg2, arg3, \dots, callback)
```

[3] See http://github.com/extjs/Connect

The callback has the following form:

```
callback(err, result1, result2, \dots)
```

By convention, `err` is `null` on success or contains an error object on failure.

Programs relying on nested anonymous callback functions can be hard to read. Therefore patterns like *continuables* or *promises* have been created to deal with this complexity (See [4] for an overview of available alternatives for node).

These patterns allow both parallel and sequential execution of asynchronous functions. At present, it is not clear how these patterns influence asynchronous TMAPI. It is notable that for some TMAPI calls the order of execution matters (e.g. as it may result in merging of certain topics), while other TMAPI calls can be executed in arbitrary order (e.g. retrieving a list of topics by subject identifier). This needs to be investigated in a different paper, and the resulting API should be evaluated through a real-world use case.

## 6 Conclusions and further work

This paper has shown that tmjs in a node environment provides an easy way of creating server applications based on Topic Maps. The PSI server application allows Topic Maps authors to publish subject identifiers from arbitrary topic maps.

Node makes it easy to create REST-based server applications or other specialized servers for Topic Maps protocols like TMRAP or TMIP.

An asynchronous interface to Topic Maps seems useful. However, further research has to be done to investigate the full potential of an adaption of TMAPI to the asynchronous execution model of node. One drawback is that this might lead to two variants of the tmjs API, one node-based API and an API for Web browsers.

## References

1. tmjs. Open-Source Topic Maps engine. http://github.com/jansc/tmjs
2. node.js. Event-driven Server-side JavaScript. http://nodejs.org
3. TM4Jscript. Open-Source Topic Maps engine. http://tm4jscript.sourceforge.net
4. Tim Caswell: async_experiments.js https://gist.github.com/602efd6a0d24b77fda36
5. Douglas Crockford. JavaScript: The World's Most Misunderstood Programming Language. 2001. http://javascript.crockford.com/javascript.html
6. ECMA International, Standard ECMA-262. 1999
7. Douglas Crockford: Javascript: The Good Parts. O'Reilly Media, 2008. – ISBN 0596517742
8. J. Schmidt: PHPTMAPI. http://phptmapi.sourceforge.net/
9. Mozilla Rhino. Open-Source Java implementation of JavaScript. http://www.mozilla.org/rhino
10. Mozilla Spidermonkey. Open-Source C implementation of JavaScript. http://www.mozilla.org/js/spidermonkey/
11. L. Heuer, J. Schmidt: TMAPI 2.0. In: 4th International Conference on Topic Maps Research and Applications. Leipziger Beiträge zur Informatik. Leipzig (2008) 129–136
12. TMAPI.Net. http://tmapinet.sourceforge.net

13. 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
14. Google V8 JavaScript Engine. Open-Source C++ implementation of JavaScript. http://code.google.com/p/v8/

# Part IX

# Topic Maps in the Industry

# Topic Maps for Subject-Centric Publishing from Document-Centric Content Management Systems – a Case Study on a Website of a Regional Cluster of Companies

Gerhard E. Weber, Ralf Eilbracht, and Stefan Kesberg

Nexxor GmbH, Vollmoellerstr. 11,
70563 Stuttgart, Germany
\{gerhard.weber,ralf.eilbracht,stefan.kesberg\}@nexxor.de

**Abstract.** Content management systems (CMS) based on document-centric information architectures come with high costs of linking content between documents. Consequently, websites published from such systems provide these links in low numbers only, leaving users with the daunting task of connecting the dots between subjects represented as isolated text fragments in different documents. With the commercial use case of a website publishing member information of a regional cluster of companies, we demonstrate how a Topic Maps-based web frontend on top of a CMS upgrades content to a subject-centric network of knowledge models, and thus multiplies the numbers of access paths, of defined, navigable, and retrievable associations between represented subjects, as well as of available views. This added value comes without added effort for manually editing and maintaining content, and – for web sites hitherto relying on manual link creation and maintenance – even reduces this cost factor.

## 1 Introduction

Whether based on a relational store or not, for many document-centric content management systems (CMS) the web page is the core structure for organizing content, and static menus dominate their hierarchical access structures. Although documents may easily be mapped into XML trees, this does not solve the problem of organizing arbitrarily structured information, as Barta [1] explained. Neither does it answer the fundamental question of what content is about, nor solve the problem of linking a subject represented within one document or web page with other relevant statements about the same subject made in other documents. Hence, editors using document-centric CMS need to command up-to-date knowledge of all content in order to be able to manually code links between different documents. This precondition for comprehensive linking of represented subjects is illusory for any system holding content of some size and dynamics. Furthermore, the time required for manual link editing and maintenance is substantial to a degree, that many websites published from document-centric CMS are characterized by a low number of such links or even their complete absence. In addition, document-centric CMS are burdened with another limitation: due to their static page structures, they do not allow flexible integration of information from additional resources.

For web publishing, one of the major uses of Topic Maps [2], the benefits of upgrading data in relational databases to a subject-centric information architecture and publishing paradigm has been demonstrated for the use case of Germany's descriptive variety lists [3]. Equally, Topic Maps offers a solution for improving navigation and use of content in document-centric CMS, as well as its enrichment with additional information. We demonstrate this claim with a commercial use case of a website publishing member data of a regional cluster of companies.

The major aim for the use case was improved support for answering domain specific questions, many of which address recognition of relationships between represented subjects. Additionally, supplementary information on each member's fields of activity stored outside of the CMS in a spreadsheet file was to be integrated. Also, improved handling of spatial relationships was to be achieved. The customer required continuous use of his CMS, and excluded any changes to his processes of content editing.

We briefly describe data and the original web frontend, and highlight its limitations with a number of relevant questions. Based on a web service delivering XML, and on a data processing and transformation pipeline, all content is transformed into a topic map, and periodically updated. The web application rendering access to the topic map is based on a commercial software package for processing topic maps, and is described in some detail.

## 2 Original Data and Web Application

Original data describes well over 200 organizations such as companies and research institutes in the biotech and medtech domain in the metropolitan area of Stuttgart and some regions to its south. Each organization is characterized with up to 20 kinds of properties such as name, address, focus, profile, management, advisory board, year founded, number of employees, or contact person. Organizations, locations, fields of activity, and years were modeled as topics. Most other attributes were modeled as occurrences.

The original web application provided one web page per organization, and a single alphanumerically sorted overview listing each organization's name and focus. The web pages for individual companies provided no links to other content relevant for the subjects related to a particular organization. Thus, users were left with the daunting task of connecting the dots between subjects represented as isolated text fragments in different documents. Some exemplary questions of considerable relevance for members as well as for the company in charge of cluster support are listed in Table 1. These questions were practically not answerable with the original web application. Full text search was no help at all for tackling these questions.

## 3 Topic Map Creation

The operative content management system was supplemented by a web service delivering required content as XML. Subsequently to content analysis of the XML string delivered by the web service, and the spreadsheet provided by the customer, an application specific Topic Maps ontology was created. Topic types included concepts such as organization,

**Table 1.** Some questions practically not answerable with the old web application

| No. | Question |
|---|---|
| 1 | Organizations operative in a particular field of activity? |
| 2 | Medtech companies headquartered in a particular city/county? |
| 3 | Biotech companies engaged in diagnostics, and headquartered in a particular county? |
| 4 | Organizations founded prior to the year 2005 and related to nutrigenomics? |
| 5 | Cities or other administrative units represented in the cluster, and by which companies? |
| 6 | Organizations were founded in a particular year? |
| 7 | Age distribution of biotech or medtech companies and research institutes in the cluster? |
| 8 | Focus on particular company: Other participants of the same projects? |
| 9 | Focus on particular company: Other companies headquartered in the same city, or administrative unit? |
| 10 | Focus on particular company: Other companies with the same fields of activity? |

location, field of activity or calendar year. Occurrence types defined included focus, profile, management or contact. Binary association types were defined for linking organizations with their headquarters (located in), their fields of activity (has field of activity), or their founding year (was founded in). In total the employed Topic Maps ontology contains 18 topic types, eight association types, 17 role types and 14 occurrence types.

The XML string delivered by the web service is subjected to an automated preprocessing routine in order to enable subject-centric mappings. In particular, we extracted e-mail-addresses, URLs, and headquarter locations from address data provided in a single text string for each member organization. For the organizations' headquarters, we applied entity recognition to address data, and resolved locations at the community level. In order to account for name variations of locations, we created a topic map for location names, assigning all occurring names to their respective location.

Based on the application ontology and the structure of the data sources, mapping rules were defined in XSLT for transforming content of the XML string and of the spreadsheet into topic maps [4] coded in XTM 2.0 [5]. Accordingly, the spreadsheet data containing the member organizations' fields of activity were transformed into a topic map.

The community level was supplemented with additional information on hierarchical structures such as county, and higher administrative units. To that end, the topic map transformation of the XML string is merged with a topic map on all relevant regional administrative structures of Baden-Württemberg.

In the final step, all topic maps resulting from the data processing stream are merged to a single topic map as the backend of the web application. The total number of topics, associations and occurrences in the resulting topic maps is about 500, 1.600, and 1.900 respectively.

Due to the limited dynamics of membership data, updates are run only once a month. For this, the data processing pipeline is started at a fixed date every month. Unless new

locations with non-official name variations have been entered in the CMS, no manual input is required for the transformation pipeline.

## 4 Topic Map-based Web Application

We used topicWorks, a commercial software package, for merging, and processing the topic maps, and for creating the publicly available web application. Based on a generic Topic Maps browser the frontend renders most interface structures from the processed topic map. Configurable interface structures and a template system allow customized views. Templates were adjusted to the customer's corporate design, and view customization was used for ontology topics such as organization, location, or field of activity.

The generic browser distinguishes two major visualization patterns. For topic types, tables of instances and their statements are generated by default, based on the graph structure of the underlying topic map. For non-typing topics such as a particular company, individual views are generated, collocating all statements on the focal topic. Templates and configuration allow for the modification of default visualization patterns, and any required view may be created. To that end, tolog [6] queries were used for supplementing additional attributes in automatically rendered instance tables, or for adding tables in individual views.

Due to the two default patterns, the generic Topic Maps browser is a web application out-of-the-box, even without any configuration at all. However, the application's utility is increased by configured views, such as member data retrieved by configured tolog queries using inference rules on the geopolitical containment associations. In this way, overviews of members headquartered at a particular location are not only available at the community level where the associations between an organization and its headquarter is defined, but also at all higher administrative levels, such as county.

In contrast to the single overview provided by the old web interface, the Topic Maps-based application renders about 500 subject-centric relational views, many of which are not customized but generated automatically due to generic visualization patterns. Table views offer controls for filtering, adding or hiding columns, and for sorting. Views on individual organizations are automatically complemented with links for all represented subjects. Each of the exemplary questions listed in Table 1. can be answered with a few clicks or a semantic search.

Figure 1 shows a screen shot of the overview on a particular field of activity, addressing question No. 1 in Table 1. The view displayed is one click away from any webpage containing the topic which represents the very field activity. Alternatively a semantic search for the field of activity would also retrieve the respective topic.

Question No. 2 in Table 1 may be answered with a semantic search for the county of interested. Figure 2 shows the respective screen shot of the search results page. The user has reduced the displayed result set to medtech companies only. The table of results also displays the type of each result, and gives the statements causing the hits. In the example, the search algorithm evaluated an inference rule on headquarter location. None of the retrieved Medtech companies is "tagged" with the county of interest, but their headquarter locations are contained in it.

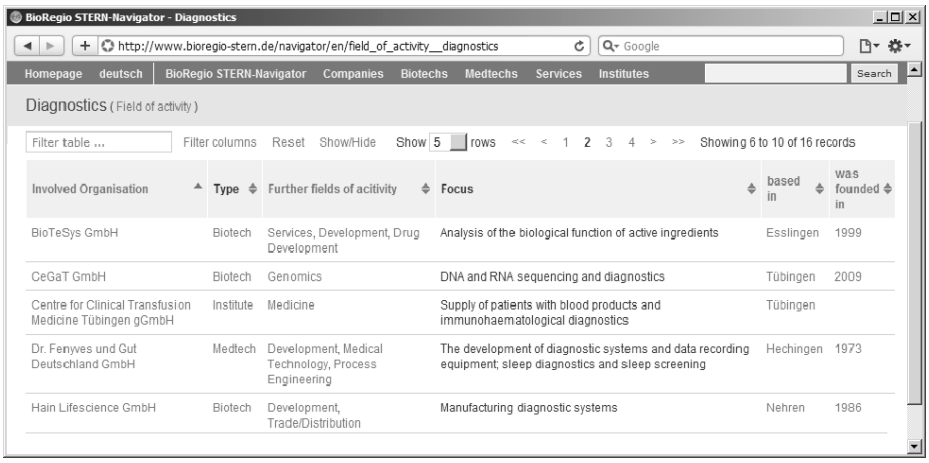



**Fig. 1.** Screenshot of the Topic Maps web frontend with an overview on organizations involved in a particular field of activity with a selection of properties; note that except for "Focus", all content is provided by topics, each of which offers access to related information

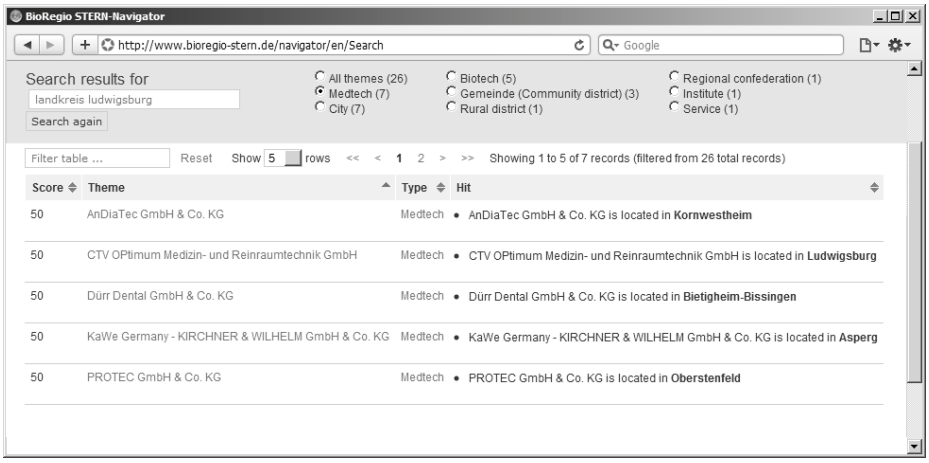



**Fig. 2.** Screenshot of the Topic Maps web frontend showing a results page for a semantic search for a particular county; displayed results were restricted to medtech companies

We did not perform a survey of user experiences. However, informal feedback from the customer's employees using the application was unequivocally positive, and the same holds for users employed by member organizations of the cluster. In particular, the ease of access to content, and the transparency of retrieving thematically related content were welcomed. Also, the qualitative difference of search results was perceived as an improvement.

The only criticism voiced by one user was the lack of faceted search functionality. Nevertheless, faceted search is indeed available for any relational view rendered by the

Topic Maps-frontend. Hence, this criticism obviously highlights a weakness in the design of the user interface for this particular functionality rather than a shortcoming of its information architecture.

## 5 Discussion

With the commercial use case of member information published by a regional cluster of companies, we demonstrated how content managed in a document-centric CMS can be published in a subject-centric way. By using a minimal domain ontology and a mapping process subsequent to some data preprocessing on XML-strings retrieved via a web service of the CMS, all relevant content is transformed into a topic map. Based on commercial Topic Maps software, a subject-centric web application adjusted to the customer's corporate design was created with some views configured by type specific templates and tolog queries.

We employed a subject-centric publication system on top of an operative document-centric CMS. To that end, a data processing and transformation pipeline was implemented, which periodically retrieves content from the CMS and a spreadsheet, and integrates it with topic maps holding name variations of locations and geopolitical containment associations of the region of interest. Whereas, this approach is feasible and has been successfully operative, other solutions for integrating subject-oriented and document-oriented management systems should not be overlooked. Of particular relevance are integrations at the tool level such as described by Garshol and Fischer [7]. In our case, this approach was not an option, since the customer preferred avoiding any changes to his operative systems, or his established editing routines.

The old website rendered from the document-centric CMS offered a single overview consisting of an alphanumerically sorted list of organizations, and member pages bare of any links to related content. By upgrading content to a topic map, data was transformed into a navigable network of associated subjects. With more than 500 overviews, and individual member pages publishing topics instead of isolated text fragments, answers to questions hitherto unanswerable can now be retrieved at the cost of a few clicks or a semantic search.

The former web application could have been improved considerably also by other approaches such as relational views on database tables or XML transformations. However, this kind of relational views need to be formulated either at design time of an application, or subsequently by someone in the IT-department. Given the graph structure of content, all relevant views will only rarely be formulated at design time, since this involves pre-considering all relevant questions users would want to ask. In addition, coding queries and configuring views of the result sets is a costly process, restricted by usually tight budget constraints. With the majority of users coping without much support of IT-departments, upgrading data to navigable graph structured knowledge models, may very well be the more efficient means for empowering users. All the more so, when visualization patterns include relational views automatically generated from the underlying graph structures.

Facetted filtering, a feature offered by many CMS, does not require a subject-centric content model, and allows answering many of the critical questions listed in Table 1. However, this kind of information retrieval requires precise knowledge of information

needs on behalf of the user herself, and additionally considerable expertise in the content domain. For a website publishing information on Bavarian companies, web analytics revealed that the vast majority of users does not at all apply this means of information retrieval. Therefore, we suppose that for our use case, subject-centric associative access paths may be of greater utility to the majority of users.

## 6 Conclusion

With Topic Maps, subject-centric web publication from content in document-centric CMS is easily achievable. Thus, added value due to improved content usability is available even without any increase of efforts or additional cost in the manual editing process. To the contrary, where content in document-centric CMS is subject to manual link editing and maintenance, upgrading to a Topic Maps-based subject-centric information architecture and publishing paradigm may yield considerable reduction of efforts and costs for the editing process.

## Acknowledgements

We would like to thank the three anonymous reviewers, whose comments helped improving this manuscript.

## References

1. Barta R, 2007. Knowledge-Oriented Middleware Using Topic Maps. Maicher L, Garshol LM (eds.) Scaling Topic Maps – Third International Conference of Topic Maps Research and Applications, TMRA 2007, Leipzig, Germany, October 2007, Revised Selected Papers. Springer, Berlin, 98–115
2. Pepper S, 2010. Topic Maps. In: Encyclopedia of Library and Information Science. Third Edition. Taylor & Francis, 5247–5259
3. Weber GE, Eilbracht R, Kesberg S, 2010. Topic Maps for improved access to and use of content in relational databases – a case study on the descriptive variety lists of Germany's Bundessortenamt. Maicher L, Garshol LM (eds.) Proceedings of TMRA 2010 – International Conference on Topic Maps Research and Applications, Leipzig, Germany, September 2010. Leipziger Beiträge zur Informatik (in press)
4. ISO/IEC 13250-2: Information Technology – Topic Maps – Part 2: Data Model. International Organization for Standardization, 2006
5. ISO/IEC 13250-3: Information Technology – Topic Maps – Part 3: XML Syntax. International Organization for Standardization, 2007
6. Garshol LM, 2006. tolog – A Topic Maps Query Language. Maicher L, Park J (eds.) Charting the Topic Maps Research and Applications Landscape – First International Workshop on Topic Map Research and Applications, TMRA 2005, Leipzig, Germany, October 6–7, 2005. Springer, Berlin, 183–196
7. Garshol LM, Fischer M, 2010. Ontopia/Liferay Integration. Maicher L, Garshol LM (eds.) Proceedings of TMRA 2010 – International Conference on Topic Maps Research and Applications, Leipzig, Germany, September 2010. Leipziger Beiträge zur Informatik (in press)

# Demo of an Automatic Semantic Interpretation of Unstructured Data for Knowledge Management

Jörg Wurzer

iQser AG, Chlupfgasse 2, 8303 Bassersdorf, Switzerland
joerg.wurzer@iqser.net

**Abstract.** The demo shows an automatic semantic analysis of Wikipedia articles about astronomy to demonstrate how to get knowledge out of unstructured data and how to access information in a new efficient way by using a concept graph and an object graph. These two graphs are the basis for further reasoning, inferring, classification and information delivery.

## 1 Introduction

The demo application shows the analytic part of a semantic middleware as a platform to build enterprise applications integrating a variety of data sources for a overall information retrieval and knowledge management as well as an overall process management.

The application is an example to demonstrate how the analytic works and how the result looks like. The user interface and the visualization of the concept and object graph could be implemented in many variations, for example in a graphical two or three dimensional net.

The research background is a fundamental research on semantic analysis including combining ideas of text mining, semantic web, epistemology and philosophy of language. The motivation is a new paradigm of accessing and organizing information for more efficiency and transparency of knowledge.

### 1.1 A new paradigm to access information

Today people use to get information by keyword search or navigating in a fixed hierarchical structure. Most document management systems and information portals are organized in that way.

In a new paradigm people do not need search for required information any more. A system delivers the required information automatically in the given context of the user. This context may be defined by an assigned task, an involved project, a customer request or a document, that has to be studied. Therefore a net of all available content objects is created by the system by an automatic semantic analysis, stored in an object graph.

### 1.2 Inferring knowledge out of a given number of data sources

One aspect of knowledge is connecting information for new statements. A second aspect is the abstraction by defining concepts and their relations in an ontology. For both aspects

the automatic analytic will support the user a semantic application. Therefore a concept and object graph is used.

The user can see whether any content object representing an instance of a concept like a Wikipedia article is related to any other content object and why there is a relation. The user can also see that the main concepts are mentioned in the data source and how they are related to each other.

## 2 Technological base of the demo

The demo is build on the iQser GIN Platform, a middleware for semantic applications. This middleware is modular system with interfaces for client systems, data sources, business process management and analytics.

### 2.1 Semantic data integration

The demo uses a part of the english Wikipedia related to astronomy. They are integrated in the system by a so called content provider. A content provider transfers content objects of a data source into a generic object, that can be processed by the system. The object is semantically typed, to let the system know to which object type the instance belongs. In the demo case we have only instances of Wikipedia articles.

Each object is described by named attributes filled by the content provider. In case of the demo one attribute is the title of the article that was found inside the extracted text from the HTML document. Another attribute is the main text of the article. Meta tags and file information can also be used for additional attributes.

A crawler is collecting the articles and looking for updates regularly. No data is imported to avoid a redundant data storage. Instead of importing the data is indexed and analyzed.

### 2.2 Automatic analysis for an object graph

After indexing the transferred objects the system starts the analysis process in three steps. In the first step the system considers the so called key attributes which are meaningful for relations to other content objects. In the demo the title was used as a key attribute. In the second step the system is analyzing the complete text in the article to identify a set of significant concepts for a similarity comparison with concepts of other content objects. In a third step the system analyses tracked activities of the user to identify repeating patterns to create new relations of modify the weight of a connection.

The three analysis methods are reflected by the description of a connection by a predicate, for example the attribute of a content object that is found in a related object. Therefore the connection can be interpreted as a subject-predicate-object-statement.

### 2.3 Automatic analysis for a concept graph

This analysis transfers the content object in a list of concepts after an iteration of all attributes and tokenizing the text values. The system calculates for each concept a value

for its significance. For the most significant concepts in a content object the system also calculated the cooccurrences. These two aspects are the basis to create a concept graph which can be visualized and queried in various way. The demo calculates a hierarchical concept tree beginning with the most significant concepts with no cooccurrences between them.

### 2.4 Related research

The demo is attached to fundamental research that has been introduced on conferences for semantic technology[1] and topic maps[2] last year. The first one describes an approach to query the object graph to answer complex questions. The second one describes the analysis as an alternative for hand crafted topic maps.

The research wants to extend the approach of Semantic Web, that is defined by the standards of W3C. The extension has three aspects:

The extension of the data sources. With the middleware any data source can be integrated as well as well as well defined triples in RDF, RDFS or OWL.

The statements in the semantic web are created automatically. It isn‘t necessary to have well predefined triples.

The statements are usually formalized in triples. The middleware uses quadruples to add a number value to describe the weight of the connection.

## 3 What the application demonstrates and the visitor can learn

The application is a combination of web client using ajax technology and the middleware of iQser for integrating and analyzing Wikipedia articles.

### 3.1 Example of a concept graph visualized as a concept tree

The visitor of the demo gets an overview over the main topics inside the collection of articles by a concept tree with four levels. Each main topic is connected to related concepts. He can follow the connection down to the fourth level. In the screenshot the visitor can see that “star” is one of the main concepts in the topic range “astronomy”. He further can see, that one aspect of the concept “star” is “observation”. In context of “star” and “observation” one concept is “instruments”. There are instruments to observe stars. In context of the three concepts “star”, “observation” and “instruments” the tree shows “spectrograph” as one example for an instrument to observe stars.

By selecting one path in the tree the visitor of the demo can see clustered and ranked articles which are related to the concept constellation. The top articles are plausible

[1] Jörg Wurzer, New approach for semantic web by automatic semantics, European Semantic Technology Conference (ESTC), Vienna 2008, http://www.estc2008.com/index.php/program/program-list/60-joerg-wurzer

[2] Jörg Wurzer, Stefan Smolnik, Towards an automatic semantic integration of information, in: Lutz Maicher Lars Marius Garshol (Eds.), Suject centric computing. Fourth International Conference on Topic Maps Research and Applications (TMRA), Leipzig 2008

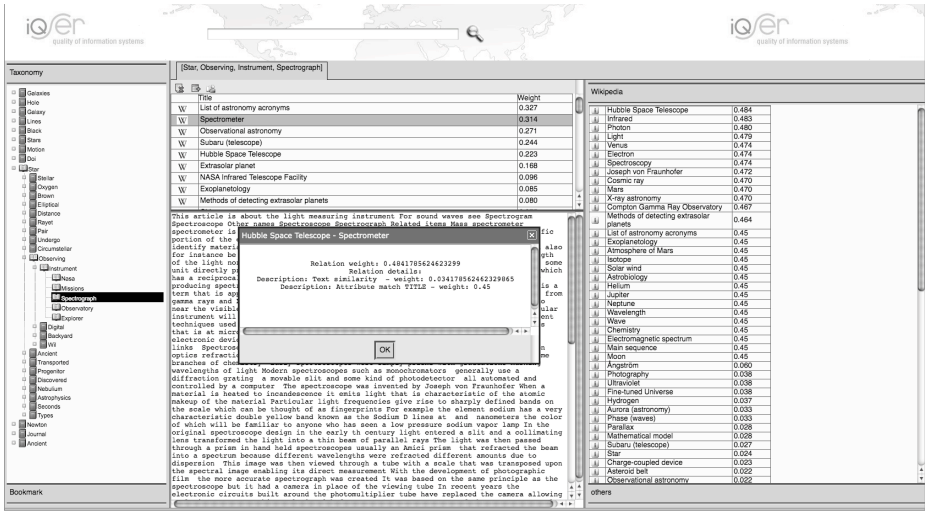



**Fig. 1.** The screenshot shows on the left site a selected path in a concept tree and the clustered articles belonging to the constellation of concepts included in the path. On the right site a ranked list of articles are shown, which are connected to an article selected in the middle of the screen.

allocations like “Spectrometer” and “Observational Astronomy”. The algorithm for clustering is combined with an algorithm for a ranking that helps the user of the application to get a well organized overview.

### 3.2 Example of an object graph visualized as list of connected Wikipedia articles

If the user of the application selects one article in the calculated cluster, then the extracted text from the HTML document of Wikipedia appears and on the right site a list of connected wiki articles. If there would be more than one object type, they would be shown in separated accordion area for a better orientation.

The related articles are ranked because of the calculated weight as a result from the combination of the weight of each found relations between two objects that is stored in an quadruple. A button at the left side of each article in the list opens a popup window with information about the reasons for the connection. They can be displayed in different ways. The demo shows just a summary but it is possible to show more detailed information like the matched concepts in a similarity calculation.

Because of the non hierarchical structure the user of the application can use the connected articles to navigate through the article collection. If he would select one linked article, this article would appear in the center of the screen and at the right site its connections including the link back to the article he selected before.

### 3.3 Combination of classical and new information retrieval

The demo offers a search field in the top of the screen. That can be used for a classical entry for research. If the user for example searches for “jupiter” that he will find a list

of articles with the article about “Jupiter” as the first hit. Now he can use this article to start a research without any further queries with a combination of keywords. He just looksWikipedia at the connection to get an overview over the related topics and opportunities to get deepening information.

The new paradigm of information retrieval is to get all relevant information in a given context. In the demo the context is an Wikipedia article. In use cases for business a context could be any other business object characterizing the current focus of interest.

# Part X

# Report from the Sessions

# Report from the Open Space Sessions of TMRA 2009

Lars Marius Grashol[1] and Lutz Maicher[2]

[1] Bouvet ASA, Oslo, Norway
larsga@bouvet.no, http://www.bouvet.no
[2] Topic Maps Lab, Leipzig,
University of Leipzig, Johannisgasse 26, 04103 Leipzig
maicher@informatik.uni-leipzig.de, http://www.topicmapslab.de

**Abstract.** Abstract.This is a summary of the presentations made in the two open space sessions at the TMRA 2009 conference. The open space sessions were free-form sessions where anyone could sign up to do a short presentation. The result is a collection of reports on works in progress and interesting ideas, some of which are likely to appear as papers at next year's conference.

## Introduction

The following is a summary of the open space sessions at the TMRA 2009 conference based on the slides used by the presenters, the videos of the presentations and impressions from the conference itself. The contributions were informal and non-refereed, since workshop attendees had been given the opportunity to sign up to short talks on a flip chart during the conference, and the suggested format for each presentation was five minutes presentation, and five minutes discussion. Both sessions were moderated by Lars Marius Garshol. The outcome of this "playground for visionaries" is this report on forward-looking work in progress, written in September 2010. The slides and the videos of each presentation are available at the conference website, and the links are provided in each presentation summary below. Besides the presentations documented below, there was one conference announcement for Topic Maps Japan 2010[1] by Motomu Naito.

## Paraconsistent Reasoning in Ontopedia

In this presentation[2] Dmitry Bogachev (Ontopedia) describes the problem that in large-scale systems traditional logic can not be used for inferencing due to the ever-present inconsistencies. As answer he proposes paraconsistent logic, especially the direct logic approach, as introduced by Carl Hewitt. Paraconsistent reasoning allows assertions to be collected from various sources new assertions to be safely inferred from them. Dmitry Bogachev presents Ontopedia as a PSI server, and as an inconsistency tolerant system of assertions populated by users. Each assertion can have multiple proposals from different sources with different truth values (monotonic false, default false, unknown, default true,

[1] http://www.topicmapslab.de/events/TMJP_2010
[2] http://tmra.de/2009/talks/Paraconsistent_Reasoning_in_Ontopedia

monotonic true). These proposals can be provided by people, external systems, or even inference modules. Furthermore the concept of contradiction levels (no contradictions, default contradictions, monotonic contradictions) is introduced. The systems should try to minimize the contradiction level for each assertion. The decision procedure within the engine tries to calculate truth values for assertions based on existing proposals and it calculates contradiction levels. The result of these decision procedures are the "visible assertions". What makes paraconsistent reasoning flexible is that new proposals can always change the truth value of an existing assertion. Any resulting contradiction does not participate in future inferences and the engine can suppress some previous inferences.

## Dynamic Integration of RDF and Topic Maps

In this presentation[3] Arnim Bleier (Topic Maps Lab, University of Leipzig) introduces SesameTM. It is a TMAPI 2.0 implementation for triple stores based on the TMDM ontology proposed by Anne Cregan[4]. He claims that it enables domain specific RDF data to be reused in a Topic Maps environment, or, alternatively, to use triple stores as a generic persistence layer for Topic Maps. The goal is to for SesameTM to support both kind of data through TMAPI interfaces. In the meanwhile SesameTM[5] has been developed to a library which is used in the Topic Maps Explorer Maiana[6] in production mode. With Nikuna the project also provides a read-only Sesame Sail implementation to enable RDF views on TMDM topic maps. For example it is used to make Maiana a Linked Data provider and a SPARQL endpoint for each public topic map.

## Temporal Qualification in Topic Maps

In this presentation[7] Rani Pinchuk (Space Applications Services) continues the discussion after the TMRA 2009 presentation from Christoph Teichmann about Temporal Qualifications in Topic Maps. He sketches a scenario about a train station with 10 platforms and several bus stations. Within this transportation system hub a huge amount of events can be observed, like incoming and outgoing trains, but also people wearing specific baggages or cleaning the platforms. Furthermore there is meta data like the official time tables. All these events has a defined reference in time. Because the sketched scenario might be not the exception but the normal case for Topic Maps-based systems Rani Pinchuk opens the discussion whether time should become a first class concept in the TMDM. His presentation[8] at the Topic Maps 2010 conference in Oslo showed that Space Applications Services already tends to think that the answer should be "yes".

[3] http://tmra.de/2009/talks/Dynamic_Integration_of_RDF_and_Topic_Maps
[4] http://www.cse.unsw.edu.au/~annec/index2.html
[5] http://code.google.com/p/sesametm/
[6] http://maiana.topicmapslab.de/
[7] http://tmra.de/2009/talks/Temporal_Qualification_in_Topic_Maps_Discussion
[8] http://www.topicmapslab.de/publications/question_answering_over_topic_maps_in_video_surveillance_application

## Some observations on types

In this presentation[9] Lars Marius Garshol (Bouvet) presents two observations on types, and asked the audience to assess whether they are trivial or not. His premise is that types are the foundation of everything in IT, like the classes in object-oriented programming, UML, and RDF, and of course the topic types in Topic Maps. His first observation is that types appear to be built into natural language. Specifically, the notion of class seems to correspond closely to common nouns in natural languages. Most of the nouns, like car, tree or house, are good candidates for types in IT. To decide whether a concept y is a type he proposes a simple language game: it is the case when you will find some x to express sentences "x is a y", otherwise y is not a type candidate. His second observation is about hierarchies. He realized that in the most cases we talk about types or classes in hierarchies, and that even small children are aware of this. For example, a child will often learn a single term (say, "woof woof") for all animals first. Later, it may distinguish between "quack quack" (all birds) and "woof wood" (all other animals). And later still, of course, terms start corresponding to species, so that "quack quack" becomes duck, "woof woof" dog, and so on. He claims that both observations worth observing for ontology modelling approaches.

## Faceted Navigation with Topic Maps

In this presentation[10] Geir Ove Grønmo (Bouvet) presents a faceted navigation plugin for Ontopia[11]. Faceted navigation is a technique for refining search results. Basically it allows the users to drill drown lists based on predefined filters, the facets. In commercial websites like product search engines faceted navigation is quite common and expected by the users. Geir Ove Grønmo presents a prototype for the website of the City of Bergen. In his demonstration he searches for Peter to get five hits in the topic map. Immediately the list of facets updates including the number of search hits behind. In the background Apache Solr[12] is used for indexing and tolog for facet definition. The source code of the implementation is available in the Ontopia project[13].

## Building Distributed Topic Maps

In this presentation[14] Andrew Townley introduces an idea about a mechanism to build distributed topic maps. His premise is that in the well established world of http and hypermedia the back-links are lost today. To avoid this situation the principles of track-backs and ping-backs are established. Andrew Townley imagines proxies (which might be semantic layers on any kind of information systems, but also materialized topic maps)

[9] http://tmra.de/2009/talks/Some_observations_on_types
[10] http://tmra.de/2009/talks/Faceted_navigation_with_Topic_Maps
[11] http://www.ontopia.net/
[12] http://lucene.apache.org/solr/
[13] http://code.google.com/p/ontopia/source/browse/trunk/sandbox/solr-utils/
[14] http://tmra.de/2009/talks/Building_Distributed_Topic_Maps

which might link to resources provided by third party topic maps services. Whenever they link to these resources they should ping-back to the content providers. These providers of structured content are free to react on this ping-backs, like negotiating about delivering more content in a preferred mime type, or simply ignoring it. The presentation “Subj3ct, representation link pingback” of Graham Moore described below concretizes this idea.

## Event Based Modelling

In this presentation[15] Lutz Maicher (Topic Maps Lab, University of Leipzig) proposes to change the view on topic maps from a static view to one that is event-based. The premise of his proposal is the observation that each topic should represent all information about its subject. The main problem with this approach is that subjects change over time. Most ontologies do not reflect this, nor do the applications built on top of them. To overcome this restriction Lutz Maicher proposes to look in a more event-based fashion on the data. Events should become the key facts in the data, each topic map should be interpreted as stream of facts. As benefit the whole life cycle of each subject will be in the topic maps. In the applications “time slice” views on the data should be used. This will enable the “good old” view, like in Omnigator or Maiana, but also new, more dynamic ones. With the revision history in MaJorToM and its usage for the map feeds and subject feeds in Maiana the first implementations of this approach can be used today.

## Methodpedia.eu

In this presentation[16] Peter Koppatz (Technische Hochschule Wildau) presents outcomes from the Comble[17] project which was funded by the EC in the Lifelong Learning Program. He shows methopedia.eu which is a wiki about new methods and activities for organizing the day-to-day work. In the background there are four instances of MoinMoin wiki[18], each for a specific language. Topic Maps is used to organize the metadata for each article in the wikis. On the long term it is planned to merge all topic map fragments into a centralized meta data repository. The main benefit of this approach is increased searchability and findability. With the Methopedia Designer a second tool is presented which allows he users to plan courses by drag-and-droping the methods collected in the methopedia wiki.

## Identifiying attributes

In this presentation[19] Peter-Paul Kruijsen (Morpheus) discussed the problem of merging large-scale topic maps data with origins in external data. As main problem he claims to get the basis for merging: PSIs for all topics, without hand-coding them. The premise of

[15] http://tmra.de/2009/talks/Event_based_modelling
[16] http://tmra.de/2009/talks/Methopedia_eu
[17] http://comble-project.eu
[18] http://moinmo.in/
[19] http://tmra.de/2009/talks/identifying_attributes

his solution is that PSIs are usually based on some kind of unique identifying attributes which are in the data. Consequently he presents a solution for auto-creation of PSI based on so called "fingerprints" generated from social security numbers, codes or even topic names. These auto-generated PSIs should be applied to topics before merging. In the presented example the configuration for the generation of these fingerprints is stored in the topic map. The pros of this approach are obvious, as main con Peter-Paul Kruijsen refers to unreadable and hence uncorrectable identifiers.

## Inferred Classification

In this presentation[20] Axel Borge presents the need for inferring classification information out of the knowledge in the ontology and the data in the underlying topic maps. The presented use case is auto-completing meta data of documents. The presenter has the idea that annotating the role types might be enough to infer the missing information, and asked the audience for its opinion.

## Triggers in Topic Maps

In this presentation[21] Lars Marius Garshol (Bouvet) proposes a trigger language for Topic Maps. The solution would be an API to register triggers. Each trigger might be a simple pair of a tolog query, used as a pattern, and a tolog update statement. For every change made to the topic map the changed topic is tested against the pattern. If the pattern matches the update statement is run against the topic map.

He introduces the following scenario: "Rani wants to be able to assign user-friendly name-based item identifiers to his topics."

The pattern part of the trigger would be the following tolog query:

```
topic-name(%topic%, $TN),
  not(item-identifier(%topic%, $II))?
```

Here `%topic%` refers to the modified topic. The pattern matches if the topic has a name, but no item identifier. This ensures that we only add an item identifier if none is already present, and if the topic has received a name.

The update part of the trigger would be the following tolog statement:

```
INSERT $topic ^ $ii . FROM
topic-name(%topic%, $TN),
value($TN, $VALUE),
/* string transform to $ii */
$topic = %topic%
```

The update statement finds the name of the topic, transforms it into a suitable item identifier (we don't show how, but it's easily possible), and finally assigns that item identifier to the topic.

[20] http://tmra.de/2009/talks/inferred_classification
[21] http://tmra.de/2009/talks/triggers_in_topic_maps

An implementation of a trigger mechanisms in Ontopia would be straightforward. An event listener has to be registered with the Ontopia event API. For each change in a topic map the affected topics has to searched and the rest is just a trivial use of the tolog API. Lars Marius Garshol also sketches two general issues with his approach. The first is how to registrater triggers so that they are persistent and reliably present in all usage contexts. The second is a the general issue of how triggers should affect transaction boundaries.

## TMBrowse Protocol

In this presentation[22] Xuân Baldauf and Lutz Maicher (Topic Maps Lab, University of Leipzig) proposes a protocol to browse remote large scale topic maps. For this a client provides some subject identification information to a content provider. In consequence the provider returns a fragment with all available information about the subject. To reach the goal of linked topic maps and to allow browsing in remote sources each topic within the delivered fragment provides additional URL to be requested to get more information about the subject from the content provider .

## Visual rendering of topic maps fragments

In this presentation[23] Terje Syversen (Ravn Webveveriet) discusses the usage of occurrences as template strings. They implemented such an approach for their internal platform to smoothly add multimedia blobs and links into occurrences.

## Subj3ct, representation link pingback

In this presentation Graham Moore[24] (networkedplanet) discusses how the identity clearing house subj3ct.com can become more lightweight. One of the most interesting concepts in subj3ct.com are representations which are links to resources that contains additional information about a specific subject. (This is a centralized approach to the more information URLs which are proposed in the TMBrowse protocol presentation.) Currently subj3ct.com this representation information to be submitted as ATOM feeds or SKOS documents. Graham Moore now proposes to allow a publisher of a resource to ping the identifier repository subj3ct.com if pertaining to that resource.

## ActiveTM for TMAPI

In this presentation Benjamin Bock (Topic Maps Lab) showed how to use ActiveTM to generate code for other targets than Ruby Topic Maps. The library ActiveTM is a Topic Maps Object mapper allowing to use domain-specific objects in the application code to

[22] http://tmra.de/2009/talks/tmbrowse_protocol
[23] http://tmra.de/2009/talks/visual_rendering_of_topic_maps_fragments
[24] http://tmra.de/2009/talks/subj3ct_representation_link_pingback

directly access a Topic Maps store. With the TMAPI extension it is possible to not only generate Ruby Code but also Java Code which accesses a TMAPI-based Topic Maps store like Ontopia or TinyTiM.

## Future of Domain Names

Benjamin Bock raised the question about the dormant or infrequently updated web presences at topicmaps.org, isotopicmaps.org and topicmaps.net. These web sites have not been updated for two to nine years but are visitied by about 50 visitors each on every day. He started a call for participation and a call for comments with the goal of updating these pages more frequently to better reflect the actual status of standards and implementations and to link to the resources the community uses these days.

## Motto of TMRA 2010

In this presentation[25] Lutz Maicher (Topic Maps Lab) crowd-sourced finding the motto "Information wants to be a topic map" of the TMRA 2010 conference. The list of the other proposed mottos might be of general interest because each of the proposals reflect a current understanding of what Topic Maps are or should be:

- Mashing
- Web 3.0 (Web X.0)
- Joking about Topic Maps
- Applications
- Back to the desktop
- Just merging!
- Insight
- Insight and disclosure
- Time and space
- Shiny
- Natural Intelligence
- Visible knowledge Networks
- Empowering Humans
- Visualization
- Mapping disagreement

[25] http://tmra.de/2009/talks/motto_of_TMRA_2010

# The Contributions for the Poster Session

Lars Marius Grashol[1] and Lutz Maicher[2]

[1] Bouvet ASA, Oslo, Norway
larsga@bouvet.no, http://www.bouvet.no
[2] Topic Maps Lab, Leipzig,
University of Leipzig, Johannisgasse 26, 04103 Leipzig
maicher@informatik.uni-leipzig.de, http://www.topicmapslab.de

**Abstract.** This paper contains the abstracts of the posters that were presented at the TMRA 2010 conference.

## Scaling Biomedical Topic Maps to Billions of Associations: How to Cope With Terabytes of Data?

Benedikt Wachinger and Volker Stümpflen

MIPS / Institute for Bioinformatics and Systems Biology,
Helmholtz Zentrum München
German Research Center for Environmental Health,
Ingolstädter Landstr. 1, D-85764 Neuherberg, Germany
{benedikt.wachinger, v.stuempflen}@helmholtz-muenchen.de

In order to understand biological systems generally and multifactorial diseases specifically, it is necessary to be able to create large-scale systems biological models as quickly as possible from the huge amounts of knowledge stored in multiple relational databases and published research articles. To achieve this we had to solve two problems: First, how do we solve the data integration problem if we want to store all that knowledge in one easily accessible place to be combined efficiently? And secondly, how do we efficiently store and manage the ever increasing amount of data, currently in the range of hundreds of terabytes?

The first problem can be tackled with Topic Maps, where a simple conversion schema from a relational database has to be developed. The data increase, however, entails that the underlying storage solutions have to scale accordingly. Since traditional approaches like relational databases do not scale well or only with a huge amount of administrational work, newer technologies are able to distribute the data to clusters with arbitrary numbers of nodes. Such technologies exist in cloud-like cluster architectures where storage and computation is done on the same machine. To use this, Google initially developed a column-oriented database concept called BigTable, which is essentially a very large key-value store. Hadoop HBase is an open source implementation of this concept. We have now invented a method for the efficient storage and retrieval of Topic Maps in HBase. We have developed a column-oriented schema, able to reflect TMs efficiently in such key-value stores. At our institute we use this schema to integrate multiple biological databases from different resources in one central repository. Additionally, we have

a semantic text mining system able to extract biologically relevant relations from the mass of available biomedical texts. Currently, our largest TM consists of over 4 billion associations.

## Use of the Wii Remote for Interaction in a Topic Map-Based Learning Web Site

Shu Matsuura

Tokyo Gakugei University, Faculty of Education, Fundamental Natural Sciences,
4-1-1 Nukuikita, Koganei, Tokyo 184-8501, Japan
shumats0@gmail.com

Topic Maps is expected to be a fundamental for the future web system that will be embedded in the daily life environment. In the future web environment, more of the digital resource will be provided directly in the location of real nature or historical area. Also, stereoscopic presentation of information will be more popular to navigate multi-domain systems. This work tries to introduce a popular game console, Nintendo Wii Remote, which is easy and intuitive to use for the interactions with the computer in the above environment. The Wii Remote navigations were implemented for a constellation observation system, a stereoscopic molecular dynamics, and the stereoscopic Google Earth.

## A generic Topic Maps viewer for the iPhone

Jan Schreiber

Ravn Webveveriet AS, Postboks 2169 Grünerløkka, NO-0505 Oslo
jans@ravn.no

Despite its bad image, JavaScript is an elegant, lightweight, and highly expressive language. With HTML5 implementaitons are becoming more widespread, almost all major browsers for the desktop or mobile devices offer advanced features like a canvas or local storage. Especially mobile Web browsers are becoming more advanced, and mobile devices provide more memory capacity, HTML5 capabilities and faster JavaScript implementations. This allows developers to easily create Topic Maps applications based on the JavaScript Topic Maps Engine tmjs.

As a demo, we have created the prototype of a simple, generic topic map viewer for the iPhone. The viewer can read JTM topic maps and provides the user with different ways to find and access relevant subjects.

Additionally, the application tries to introduce a different way of viewing topic maps as other generic viewers such as the Omnigator or Maiana. It takes the focus away from association types, role types, name types and occurrence types and concentrates on topic types and on giving the user quick access to the subjects of the topic map.

In its current version, tmjs lacks some of the features required to implement permanent local storage of topic maps, so the current implementation relies on memory storage only.