

Report from the Open Space and Poster Sessions ^{*}

Lars Marius Garshol¹ and Lutz Maicher²

¹ Ontopia AS, Oslo, Norway,

larsga@ontopia.net, <http://www.ontopia.net>

² University of Leipzig, Johannsgasse 26, 04103 Leipzig,

maicher@informatik.uni-leipzig.de,

<http://www.informatik.uni-leipzig.de/~maicher/>

Abstract. This is a summary of the presentations made in the poster session and the two open space sessions at the TMRA 2006 conference. The poster session consists of peer-reviewed conference submissions accepted as posters, while 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.

1 The Poster Session

This report summarizes the six contributions to the poster session of TMRA 2006, based on the submitted abstracts, the actual posters, and impressions from the presentations. All posters were submitted to the Program Committee, refereed and accepted as poster presentation for the conference. All posters were exhibited during the whole conference and presented in a series of ten minute presentations during the poster session. The poster session was moderated by Steve Newcomb. All posters are available online at the conference website.

1.1 TMAPI-implementation for Shared Accessed Topic Maps

Jan Hellich and Martin Krüger introduced a TMAPI implementation in Java³ that delegates to a remote server (also implemented in Java) using a web service interface. This allows multiple clients to connect to the same remote server, using the same interface as if the accessed topic map had been local. The remote server uses a persistent implementation based on a relational database. The solution uses a 3-tier approach to divide the tasks of handling the user interface, processing the topic maps (inserting, finding and deleting information), and persistent storage.

^{*} To appear 2006 in: Maicher, Lutz & Sigel, Alexander & Garshol, Lars Marius (eds.): *Leveraging the Semantics*. Proceedings of TMRA'06 - International Workshop on Topic Map Research and Applications: Leipzig, Germany, October 11-12, 2006; LNCS 4438, Springer-Verlag, see <http://www.springeronline.com/lncs>

³ http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/POSTER_HellichKrueger.pdf

Using the solution Java applications implemented against the (Java interpretation of) TMAPI⁴ can use a shared and remote relational database. Each access or modification of information stored in the Topic Maps backend will be translated into web services communication. The requests correspond roughly to TMAPI method calls, and the application server transforms these calls into SQL. The application server used is Glassfish⁵, and the database is MySQL. The architecture of the solution allows the replacement of the application and relational database servers by any conformant product.

The implementation is only a proof of concept, but successfully passes the TMAPI test suite. Performance tests yield encouraging results.

1.2 A Tiny, Simple, and Versatile Topic Map

Motomu Naito presented a Topic Map which could be used as a common test bed⁶ within the community when new technologies emerge within the standardization or research process. The Opera Topic Map⁷ authored by Steve Pepper is commonly used for these purposes, but Naito-san argued that a new common test bed is needed which is easy to understand for many people, is applicable to various problems, and is open content.

For this purpose, he proposes a “family tree topic map”. This is a small and flexible ontology (including a modeling method [Maicher06]) for the creation of topic maps representing family relationships. The vocabulary consists of one topic type “person” and three association types “brother-sister”, “married-couple” and “parent-child” (with role types). Using these types, family relationships can be modeled. Within the contribution it is shown how the given vocabulary has to be used to model family relationships as Topic Map.

Topic Maps using the given ontology (and modeling method) can be used to discuss querying, constraints, and numerous other application issues.

1.3 PHPTMAPI

Johannes Schmidt and Andreas Stephan introduced PHPTMAPI⁸, which is an adaptation of TMAPI to PHP5. TMAPI is the common programming interface for accessing and manipulating data held in Topic Maps, defined in Java. PHPTMAPI⁹ is Open Source and allows developers to write Topic Maps-based applications in PHP against a common interface. Although PHPTMAPI is intended to be a full implementation of the TMAPI, some small derivations are introduced by the authors (ie. locators are directly represented as strings, not as locator objects).

They also introduced QuaaxTM¹⁰, a Topic Maps engine implementing PHPTMAPI. QuaaxTM uses MySQL with InnoDB as storage engine and therefore benefits from

⁴ <http://www.tmap.org>

⁵ <http://glassfish.dev.java.net/>

⁶ <http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/tmra2006-posternaito.ppt>

⁷ http://www.ontopia.net/omnigator/models/topicmap_complete.jsp?tm=opera.ltm

⁸ <http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/tmra2006-PHPTMAPI.png>

⁹ <http://phptmap.sourceforge.net>

¹⁰ <http://quaaxtm.sourceforge.net/>

transaction support and referential integrity. Version 0.2 is available (2007-01-16), but does not implement the merging mechanism of the TMDM.

PHP and MySQL are commonly used to build web applications. Full featured Topic Maps engines for these environments will be very valuable to put Topic Maps into the web.

1.4 Natural Language User Interface to Topic Maps with Prolog

Thomas Flemming presented an approach for a natural language user interface to Topic Maps¹¹. Flemming used Prolog as a backend for storing, querying, and updating Topic Maps. Generally, approaches for natural language user interfaces to Topic Maps can be used to increase the search quality and create human-centric interfaces to the database. Another issue will be the ability of the modification of topic maps using (nearly) natural language phrases.

In the first part of the contribution it is shown how Prolog can be used as backend for Topic Maps. Prolog, one of the veterans in logical programming, with a long-standing history starting in 1972, has the benefit of being a mature programming language well suited for Topic Maps. Besides the declarative part, rules can be used to define views on the backend. Even the possibility of the definition of constraints is given. There is a Java implementation of the backend.

In a second step, information given in the topic maps is mapped to some generic natural language elements, like nouns, proper nouns, verb phrases, and transitive verbs. Lexical knowledge is added to the domain models represented in the topic maps. As example the topic with the basename “Johan Sebastian Bach” is given, which is a proper name, a proper noun, and a noun phrase. Further, association types normally consist of Verb Phrases like “has composed”. This lexical knowledge can be assigned manually or automatically (using NLP techniques) to the topic map.

So called pattern of interests can be defined or identified (as example the pattern “ProperNoun VerbPhrase NounPhrase” is given) and further translated into Prolog queries (with unfilled slots). Completely filled patterns (facts) are almost natural language, so that search results can be represented in this way.

In the context of this contribution the thematically closely related project AIOBCT¹² [Sigel05] should be mentioned, where a natural language interface to Topic Maps is implemented based on Toma [Pinchuk06].

1.5 SIREN—a Topic Maps-based Semantic Information Retrieval Environment for Digital Libraries

Hendrik Thomas (Bernd Markscheffel and Torsten Brix) presented SIREN¹³, a scientific framework of a Topic Maps-based “Semantic Information Retrieval Environment for Digital Libraries”. The core of the system is a domain model expressed as a topic

¹¹ http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/POSTER_thomasfl.pdf

¹² <http://www.informatik.uni-leipzig.de/~tmra05/PRES/RP.pdf>

¹³ http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/POSTER_SIREN.pdf

map which acts as a semantic layer improving the information retrieval. SIREN is developed within the “Digital Mechanism and Gear Library” (DMG-Lib¹⁴) project.

SIREN integrates existing prototypes to improve information retrieval in the digital library using Topic Maps. The information retrieval process is seen as a four step chain, starting with the creation and population of the domain model and ending with the visualisation of the request results.

The first step is subject analysis. The first part of this step is creating a domain model (in the project context: about mechanisms and gears) as topic map. For this manual task the Topic Maps Wiki Editor TMwiki¹⁵ [Sigel05], is used. The second part of this step is assigning of relevant digital content to the subjects in the domain model. For the creation of these occurrences three indexing levels are foreseen and partly implemented: (1) automatic indexing based on statistical analysis, (2) expensive and reliable content creation by experts, and (3) flexible collaborative tagging approaches.

The second step of the chain is the specification of information needs. To express the information needs the users can exploratively browse through the domain model by using the Topic Map Visualiser (TMV), even developed by the project team. Additionally, the users can express their information needs by (simple and complex) queries or the selection of a set of topics in interest.

The third step of the chain is the search process where the information satisfying the expressed information needs have to be found. This classical information retrieval task uses both the domain model and the indexed digital content.

The last step is the visualisation of the search results. The Topic Maps Viewer presents the search result in the context expressed by the domain model.

The presented work introduced the conceptualisation of a framework for improving information retrieval in digital libraries based on Topic Maps technologies. To get a stable framework in future, the prototypes have to be developed further to stable and scalable solutions which have to be integrated on a reliable platform.

1.6 Visualisation of Intellectual Content for Topic Maps Conceptualisation

Patricia Cheryl Liebetrau presented the challenges faced by a national pilot digitization project in South Africa¹⁶: making the transition from a project-centric site to an information-centric resource. The cultural heritage project, entitled Southern African Freedom Struggles, 1950 to 1994, has been an attempt to provide a comprehensive archive of literature and online resources to support a changing higher education curriculum in South Africa and to provide an important resource for historians, political scientists and students of African Studies in a global context. The digitalized materials which have to be integrated is everything from underground journals, minutes of political meetings, ANC materials, etc. The aim of the project is to ease the access to this heterogeneous information by implementing a subject-centric knowledge organization.

Within the DISA¹⁷ project a thesaurus was developed which helps to organize apartheid related subjects from a scholar perspective. In future, this thesaurus should

¹⁴ <http://www.dmg-lib.org/dmglib/main/portal.jsp>

¹⁵ <http://www.informatik.uni-leipzig.de/tmra05/PRES/HTa.pdf>

¹⁶ <http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/tmra2006-liebetrau.jpg>

¹⁷ <http://www.disa.ukzn.ac.za/>

be the backbone of a Topic Maps-based solution for organizing the digitalized content and for visualizing the conceptual relationships between them. The poster presented conceptual work in progress.

2 The Open Space Sessions

The following is a summary of the open space sessions in chronological order based on the slides used by the presenters, a blog summary of the conference¹⁸, and impressions from the conference itself. The contributions were informal and non-refereed, since workshop attendants 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: one slide, 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.

2.1 A Topic Map for Botanical Art

Motomu Naito presented a topic map about various species of flowers that has two points of interest. Firstly, it was produced from a relational database of botanical information merged with a hand-written topic map. Secondly, the topic map was used to produce a PDF document using XSLT and XSL-FO.

The database is actually a CSV file with a standard biological classification of 386 families of flowers into 83 orders, 11 subclasses, 2 classes, and 1 phylum. The hand-written LTM file [Garshol06a] has individual species of flowers and the families to which they belong, people, dates, places, and pictures. Naito-san has created the data flow shown in 1 on the following page, converting this into a PDF document. (DB2TM is a commercial tool for integrating relational data with Topic Maps described in 2.8 on page 11.)

Once the two data sources have been merged into a single topic map that topic map is exported to TM/XML [Garshol06b]. This is an XML syntax for Topic Maps that is much easier to process with XSLT than XTM, and this is what Naito-san has done. The XSLT stylesheet produces XSL-FO output, which is again converted to PDF.

2.2 A Standard Web Service Interface for Topic Maps

Graham Moore spoke on the need for a standard web service interface to Topic Maps servers. His rationale was that although an interchange format (like XTM) allows data to be moved between applications it isn't actually sufficient to enable applications to interoperate. Given that web services are currently the most common method for accessing remote applications independently of platform having a common standard web service interface for Topic Maps servers seems the best way to achieve this.

Graham listed a number of such web services already in use:

¹⁸ <http://www.garshol.priv.no/blog/73.html>

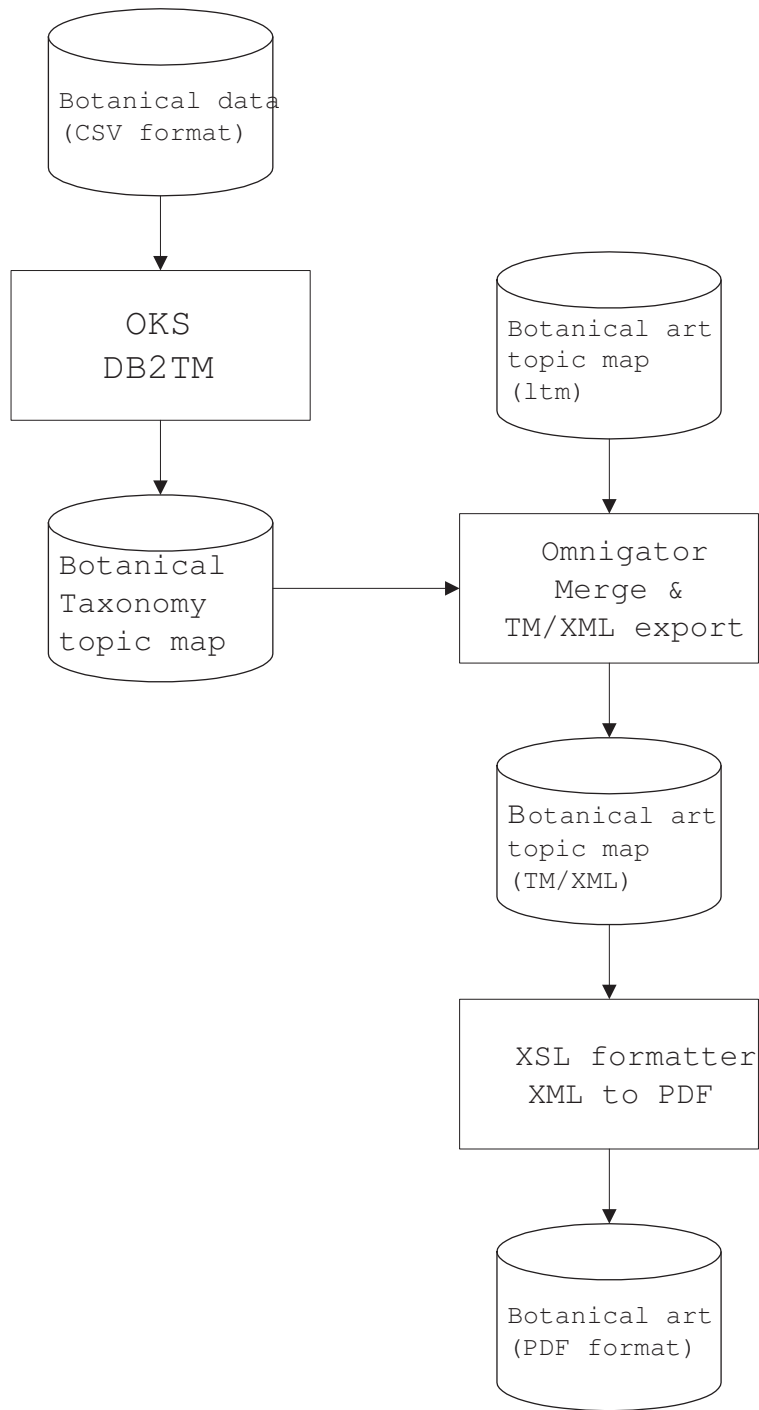


Fig. 1. Data flow

- The interface to NetworkedPlanet’s commercial TMCORE Topic Maps engine¹⁹.
- Ontopia’s TMRAP interface used in their commercial OKS product[Garshol06c].
- Robert Barta’s TMIP interface implemented in his open source Perl engine[Barta05a].
- The interface used by Robert Cerny to make Topincs[Cerny07].
- The interface used by Jan Hellich and Martin Krger for their remote TMAPI implementation (see 1.1 on page 1).

According to Graham this shows both the need for such a service, and that there is sufficient experience in the field already that a useful common standard can be created.

2.3 Proposals for Graphical Topic Maps Notations

Professor Jaeho Lee presented two proposals for graphical Topic Maps notations. ISO is currently working to create a standard for this, and professor Lee’s work was two proposals for what the standard might look like. The ISO standard will support both schema information and visualization of instance information, but professor Lee’s proposals covered only instance information.

The first proposal uses a fixed set of graphical shapes, with boxes for topics and lines for associations and occurrences. Reification is done with identifiers and references to identifiers. The shapes are shown in 4 on the next page, 2, and 3 on the next page. Professor Lee demonstrated an Eclipse plug-in which can be used to create models using this graphical notation, and expressed surprise at how easy creating the plug-in was.

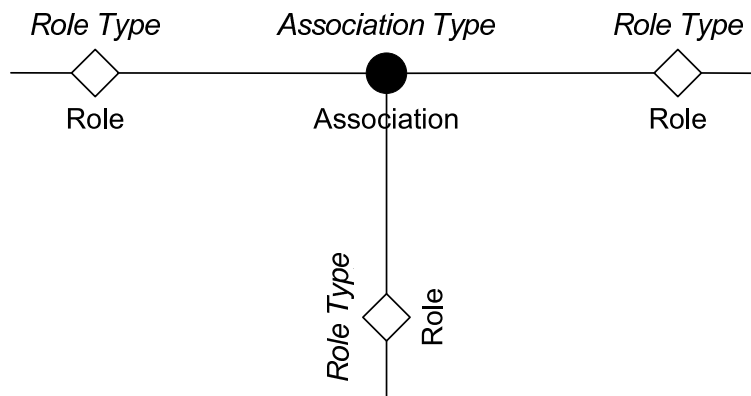


Fig. 2. Associations

The other proposal was more free-form and allowed any shape to be used for individual topics, together with a “legend”, which maps sets of shapes to the topic type they are used to represent. It was not clear how other Topic Maps constructs were represented in this notation.

¹⁹ <http://www.networkedplanet.com/technology/webservices/intro.html>

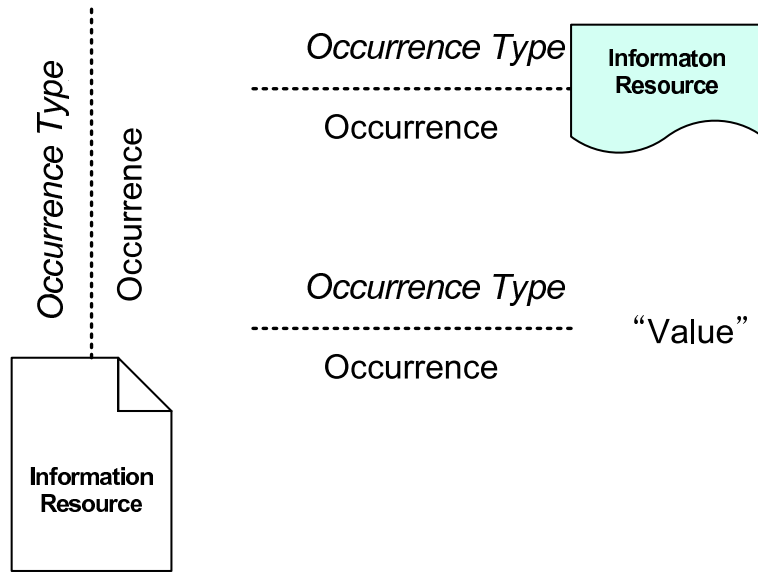


Fig. 3. Occurrences

Topic Type



Fig. 4. Topic

2.4 The Atmo Topic Maps Editor

Xuan Baldauf presented Atmo, a Topic Maps editor written in Java. Atmo does not use a schema to guide the editing, which means that when creating, for example, a binary association, the user has to choose four topics (the association type, two role types, and the topic at the other end). The difficulty with this is that without a schema any topic could potentially appear in any of these four positions, which makes it awkward for the user to choose the correct topic.

The distinguishing feature of Atmo is that it uses a probabilistic model to compute the most likely topics to be chosen in each position and presents them in order of descending probability. This greatly simplifies selection of the correct topic for the user.

The model is computed from existing data in the topic map, so that the more often a topic is used as an association type (for example), the more likely Atmo considers it as the chosen association type. Similarly, it models which association roles are used together with which association types, and what types of topics play what association role types.

2.5 A Visual Ontology Editor

Graham Moore described what he called “a visual ontology editor”, which is part of NetworkedPlanet’s commercial offering. It’s based on Microsoft Visual Studio, and has predefined shapes which can be used to model Topic Maps ontologies. It supports topic, association, and occurrence types, creating a type hierarchy, as well as cardinality constraints. It’s also possible to create PSIs for the ontology topics by defining a base URI to which all identifiers get appended to create PSIs.

A standard feature of Visual Studio is the ability to export such graphical models created from predefined shapes into an XML representation. NetworkedPlanet’s tools can then process and load this into a topic map.

2.6 Creating Topic Maps in Pidgin English

Lars Heuer presented AsTMa= 2.0[Barta05b], a compact plain-text syntax for Topic Maps, which allows Topic Maps to be written in a style that is quite close to English. There is also a more formal style, and this allows Topic Maps information to be written in two different styles. One style is close to the original AsTMa= syntax, and can look as shown below.

```
paul-mccartney
http://en.wikipedia.../Paul_McCartney
name: Paul McCartney
homepage: http://mccartney.com
shoesize: 42
date-of-birth: 1942-06-18
sings: Yesterday, all my troubles seemed...
```

This creates a topic with an identifier (first line), a PSI (second), a name, and four occurrences of different types. The datatypes of the occurrence values are automatically recognized from the syntax of the values.

However, what is more unusual is the new style, which allows topics to be written in something approaching English. (Or pidgin English, as Lars called it.)

```
paul-mccartney plays-for The-Beatles,
  which isa music-group and which
    is-located-in London,
paul-mccartney plays piano
and has shoesize 42
```

The first line is an association, where the roles are left out, because `plays-for` has been defined as a template (which is explained below). The `,` which starts a relative clause (a nested block), where the `which` refers back to the last topic mentioned before the start of the clause. The second comma ends the relative clause, and so on the fourth line we have another association. The general rule is that a new association with no `which` or `who` refers to the left-hand topic of the previous association, so on the fourth line it's `paul-mccartney` who plays the piano, and not `The-Beatles`. In the final line, the `has` indicates that what follows is an occurrence type and a value for the occurrence. The `ands` are ignored.

This means that we could rewrite the previous example to the following (note that a blank line after each association would be necessary to make this correct AsTMa=), where all topic references are given explicitly.

```
paul-mccartney plays-for The-Beatles
The-Beatles isa music-group
The-Beatles is-located-in London
paul-mccartney plays piano
paul-mccartney
shoesize: 42
```

As stated above, this works because `plays-for` is defined as a template, as follows:

```
plays-for isa astma:template
astma:body: ""
  plays-for(player: $_left, group: $_right)
""
```

The `$_left` variable is automatically bound to the value on the left of `plays-for`, and `$_right` to the value on the right of it.

It's also possible to do queries in a similar style. The following returns all topics of type `person`:

```
$who isa person?
```

The next example returns all members of the Beatles who play the drums.

```
select $who
  is-member-of(group: The-Beatles
  member: $who)
  and $who plays drums
```

2.7 A Topic Maps Module in Perl

Robert Barta presented his Perl Topic Maps engine²⁰. The current version is a revision of his earlier, more XTM-based engine. The new version is based on the TMRM, and has an implementation of TMDM on top of the TMRM. It supports AsTMA 1.x and LTM[Garshol06a]. There is also support for volatile and persistent indexes. Further, there are the beginnings of support for TSQL.

2.8 Converting Relational Data to Topic Maps

Lars Marius Garshol presented DB2TM, which is part of Ontopia's commercial OKS product²¹. This module (mentioned in Motomu Naito's presentation above) can convert relational data (via either CSV files or JDBC) to Topic Maps using a simple configuration file that defines the mapping.

A simple example of such a mapping can be seen below:

```
<relation name="organizations.csv" columns="id name url">
  <topic type="ex:organization">
    <item-identifier>#org${id}</item-identifier>
    <topic-name>${name}</topic-name>
    <occurrence type="ex:homepage">${url}</occurrence>
  </topic>
</relation>
```

This maps each row in the CSV file to a topic of type organization, and creates an item identifier for that topic based on the `id` column. A name is assigned from the `name` column, and an occurrence from the `url` column. More advanced mappings are possible, of course.

It's possible to add static data simply by leaving out the column references, and values can be transformed using either a mapping table in the XML file or a call to an external Java function. There is also support for filtering the incoming rows.

The most important functionality, however, is the ability to periodically synchronize the topic map with updates from the data sources.

2.9 Using SQL in AsTMA=

The previous presentation inspired Robert Barta to argue that instead of a dedicated syntax to import relational data into Topic Maps, one could use templates, as proposed for CTM, and one of the "include/import" mechanisms which are necessary anyway.

²⁰ <http://search.cpan.org/search?query=tm&mode=dist>

²¹ <http://www.ontopia.net/solutions/products.html>

In AsTMa 2.0[Barta05b] there is already an 'include' directive to load (and merge) external topic maps:

```
%include http://example.org/included.xtm
```

This directive can also be used with a UNIX pipe:

```
%include echo "SELECT ...." | mysql | awk '...' |
```

whereby the pipe creates AsTMa= text on the fly, which is then processed. A small extension would allow this mechanism to generate tabular content from, for example, a SQL database or an Excel spreadsheet, and then use a template to convert each incoming line into an AsTMa fragment:

```
%include 'echo "SELECT id, name, age FROM persons" |
mysql --batch company_db - |' tperson
```

This requires tperson to be defined earlier as a template:

```
tperson isa astma:template
return : ""

    {$0} # use that as topic item identifier
    name : {$1}
    age : {$2}
    desc : Welcome to corporate hell

    is-employed-at (
        employer : my_big_corporation
        employee : {$0}
    )
""
```

2.10 A Topic Maps API in Python

Lars Heuer presented Mappa²², a Topic Maps engine written in Python which has some slightly unusual features in its API. The API allows occurrence values to be retrieved and set as though the topic object were a dict (or hash), as shown in the example below. (Note that >>> is the prompt of the Python interpreter.)

```
>>> person = myMap.topicBySubjectIdentifier('http://psi.beatles.com/Paul')
>>> person['homepage']
['http://www.paulmccartney.com', 'http://www.paulmccartney.de']
>>> person['homepage @de']
['http://www.paulmccartney.de']
>>> person['shoesize']
```

²² <http://code.google.com/p/mappa/>

```
[ ]
>>> person['shoesize'] = 42
>>> person['shoesize']
[42]
```

An extra point to note here is that because the occurrence value is an integer (it's given as a Python integer value) the engine detects this and automatically assigns the correct XSD datatype URI. It's also possible to represent types which are not directly supported by setting the datatype URI explicitly, as shown below.

```
person['something'] = 'this is not a string', 'http://....'
```

References

- [Sigel05] Sigel, A.: *Report on the Open Space Sessions*. In: Maicher, L.; Park, J.: Charting the Topic Maps Research and Applications Landscape, TMRA 2005. LNAI 3873, Springer, (2006).
- [Pinchuk06] Pinchuk, R.; Aked, R.; de Orus, J. et al: *Toma - TMQL, TMCL, TMML*. In: Procs. of TMRA 2006, this volume (2007).
- [Maicher06] Maicher, L.; Bttcher, M.: *Closing the Semantic Gap in Topic Maps and OWL Ontologies with Modelling Workflow Patterns*. In: Journal of Universal Computer Science, Special Issue I-Know 2006, pp. 261-269, (2006).
- [Garshol06a] Garshol, L. M.: *The Linear Topic Map Notation–Definition and introduction, version 1.3*. Ontopia Technical Report. <http://www.ontopia.net/download/ltm.html>
- [Garshol06b] Garshol, L. M.: *TM/XML – Topic Maps fragments in XML*. In: Maicher, L.; Park, J.: Charting the Topic Maps Research and Applications Landscape, TMRA 2005. Lecture Notes in Artificial Intelligence, Springer Verlag, LNAI 3873, 2006.
- [Garshol06c] Garshol, L. M.: *TMRAP – Topic Maps Remote Access Protocol*. In: Maicher, L.; Park, J.: Charting the Topic Maps Research and Applications Landscape, TMRA 2005. Lecture Notes in Artificial Intelligence, Springer Verlag, LNAI 3873, 2006.
- [Barta05a] Barta, R.: *TMIP, A RESTful Topic Maps Interaction Protocol*; Extreme Markup 2005, Montral, Canada. <http://www.mulberrytech.com/Extreme/Proceedings/html/2005/Barta01/EML2005Barta01.html>
- [Barta05b] Barta, R.; Heuer, L.: *AsTma= 2.0 Language Definition*; Bond University, 2005. <http://astma.it.bond.edu.au/astma=-spec-2.0r1.0.dbk>
- [Cerny07] Cerny, R.: *Topincs: Topic Maps, REST and JSON*; In: Procs. of TMRA 2006, this volume (2007).