Using Knowledge Management Technology to Aid Knowledge Transfer

Alexander Zangerl

Bond University, School of Information Technology
az@bond.edu.au

Abstract. This paper describes a practical system designed to create, maintain and manage teaching materials using knowledge management techniques. This system uses ISO Topic Maps to represent and store knowledge, which subsequently are accessed online or converted into multiple forms suitable for use as lecture materials.
The benets of such a generic semantic knowledge store that represents information in a shallow and concise fashion include the ease of selecting pertinent material and of any subsequent transformation into presentations and lecture materials. Separating content lters and presentation rendering from the actual knowledge also allows for very ecient long-term maintenance of the knowledge.
This environment has evolved over the last six years and become a major factor in reducing the eorts necessary to keep course materials up to date where the subjects cover fast-changing, emerging technologies and therefore require frequent revision.
Our knowledge management approach to maintaining teaching materials also suggested itself as a novel assessment format: in advanced subjects we have had quite positive experiences with tasking students with research exercises where the deliverable is such a topic map instead of the more common essay format.
This paper gives an overview of our integrated environment and outlines the benets and positive results we have experienced with its use for teaching purposes.

1 Introduction

Working in a teaching environment where subjects covering emerging technologies are taught, the issue of *efficient and maintainable* knowledge engineering is of major importance where the creation and long-term maintenance of teaching materials is concerned: it is often necessary to introduce new material to students and therefore teaching support mechanisms need to be nimble.

Knowledge - in its various forms - has to be represented and subsequently tailored for and presented to dierent audiences. Using solely presentation technologies like PowerPoint and similar suers from their “write-only” nature, as it is quite time-consuming to keep such material up-to-date; it is also very hard to reuse the information in other contexts or combine multiple sources eciently. As we believe rmly in not reinventing the wheel unnecessarily, we developed a knowledge representation and presentation solution that focuses on reusability and simplicity.

In this paper we describe this holistic approach, where the relevant knowledge is expressed and stored in a manageable, semantically-enhanced form in one shared knowledge store, and separate from any presentation or rendering information. Once this semantic representation is created, it can be used for transformation and presentation in dierent forms ranging from slide shows to hyperlinked relationship graphs. Subsequent mod- ications or updates of the content are made to the shared knowledge store, which avoids having to maintain variant copies. Also, using a highly-structured knowledge representation allows a much more disciplined but convenient approach to linking to external resources like online articles or books.

This knowledge management approach we apply in four dierent areas related to teaching:

- as a means of representing and storing relevant information about concepts
- to create tailored presentation materials from the knowledge store
- as a source of secondary, background reading material for students
- and nally as an assessment format.

In the following, we present a brief introduction to knowledge representation with Topic Maps. This provides the foundation for the next section which covers our knowledge server's functionality as it has evolved over time, followed by the mechanisms we use to create teaching materials from this knowledge. Finally, we describe our approach to and our experiences with using Topic Map creation as an assessment format.

2 Topic Maps as Knowledge Store

Topic Maps are a knowledge representation technology belonging to the Semantic Web arena, where they compete with RDF, the Resource Description Framework. Like RDF, Topic Maps provide mechanisms to represent highly irregular data in a consistent form. Both serve for the expression of shallow knowledge and meta data and focus on representing relationships between concepts.

Both Topic Maps and RDF are standardized technologies: RDF is endorsed and favoured by the W3C, while Topic Maps are covered by the ISO/IEC 13250:2003 standard.

The primary dierences lie in the level of complexity exposed to a knowledge engineer: While RDF uses solely *subject-predicate-object* triples, Topic Maps use more exible associations to link together more heavy-weight subjects. Thus RDF triples closely resemble a directed graph of blank nodes, whereas Topic Maps can be visualized as undirected graphs linking nodes with (possibly substantial) amounts of attached data.

Somewhat paradoxically, it is the higher complexity of Topic Maps which made us prefer it over RDF: the generic triple structure of RDF is appealing in its straight-forward simplicity, but TM statements (associations) are more exible and provide higher expressivity to an author. This higher expressivity of course causes substantially higher complexity of the underlying data model, but that is not a major concern to a user.

From a lecturer's perspective, the exible statements in Topic Maps provide a very convenient way of expressing factual statements and representing annotations in textual form or by giving URLs linking to relevant sources.

Topic Maps (like RDF) support an XML-based serialization format called XTM, which serves mainly for information interchange: XML is excessively verbose for manual authoring, and therefore numerous short-hand notations have been developed to aid human authors. In the Topic Map arena no nal consolidation of these author-oriented notations has happened yet (cf. RDF and the N3 format), but conversion via XTM is always possible.

We have based our eorts on a short-hand format called AsTMa=. Because of space constraints only a short demonstration can be given here and we refer the reader to [1] for full language specication and tutorials.

Topic maps built with AsTMa= are plain text les which can be quite convenient especially for activities like tracking changes with standard tools like diff or revision control systems. As an example, consider this fragment from a map covering the history of the UNIX operating system:

```
unix-os
bn: UNIX Operating System
bn(obsolete): UNICS
in: first incarnation in 1969
dennis-ritchie (computer-scientist)
bn: Dennis Ritchie
in(interests): languages
oc: http://www.cs.bell-labs.com/who/dmr/
(developed-by)
obj: unix-os
developers: ken-thompson dennis-ritchie
developers: doug-mcilroy lotsa-others
(employed-by)
employer: bell-labs
employee: ken-thompson
period: sixties-to-y2k
```

The first two paragraphs represent two topics, starting with identiers that can be used later to refer to the topic elsewhere. Additionally, for Dennis Ritchie a categorization as computer scientist is given. Every topic can have names, ocial ones, abbreviated ones or otherwise. These so-called basenames are listed in lines starting with the keyword bn. In our case, we attach both the common name UNIX as well as the older (pun-based) name UNICS to the topic, and additionally the older name is typed as being obsolete. Another property of topics are occurrences, marked by the keyword oc; these are URIs and provide connections between the topic map and the outside world by linking relevant outside sources to this topic. The Topic Map standard also supports textual, inline, occurrences for which we use the keyword in. Both URL and inline occurrences can be of a type, such as in in(interests). This helps to qualify the relationship between the topic and the occurrence.

Such topic specications look very much akin to bookmarks on steroids and are very useful as such. Nevertheless topics are not the main vehicle for representing knowledge in Topic Maps, as the main focus is on the relationships between concepts or topics. In Topic Maps these relationships are known as *associations* and serve to connect (multiple) topics in a exible manner.

All associations have a type (developed-by and employed-by in our example) and connect two or more concepts in an explicit fashion by also including the role the particular topic is playing in the association: Ken Thompson plays the role of being an employee in that association, where Bell Labs is the employer. Dierent from RDF where triples connect exactly two subjects, Topic Maps allow us to connect an arbitrary number of topics in a single association. In our example, the statement about Ken Thompson's tenure at Bell is made more precise by including the period of 1960{2000 as a qualifying extra player.

3 Knowledge Server Technology

The evolution of our knowledge server is described in detail in [2], and this section covers only the most crucial aspects of our environment.

Our knowledge server is implemented as a web application with an additional CVS interface for external access to and maintenance of maps. For the server infrastructure open source technology was chosen and integrated: Apache provides a well-seasoned web server environment, with mod perl hosting our applications eciently. For preprocessing, templating and XML-related activities, HTML::Mason and AxKit are used. Processing of the maps themselves is handled by our own Perl modules from the public TM suite[3].

The knowledge server provides two main functions: serving of documents related to Topic Map research and developments, and providing us with the online Map Development Area. The document hosting aspects are fairly straightforward, while the Map Development Area (MDA) has become our main tool for knowledge engineering in support of our teaching.

The MDA provides a navigation interface for browsing through all our public maps, as well as editing features to registered users. Ecient and comfortable navigation through inherently unsorted elements like topics in maps requires the interactive, hyperlinked display of maps, topics and their characteristics, and to provide this we decided to present one topic per page, but hyperlinked to all related topics and associations.

The map display layout was inspired by the early versions of the TM4J project[4], a Topic Map framework for the Java language. In this layout, every topic corresponds to a single web page which covers all the aspects of the topic: directly attached characteristics like basenames or occurrences are shown centrally, whereas associations in which this topic is involved in are displayed at the perimeter. All elements that involve other topics (e.g. occurrence types, other players in associations) are shown as hyperlinks which lead to the respective topics' pages.

We put considerable eort into designing a maintainable addressing scheme for URLs on our server, which allows all relevant objects to be addressed unambiguously and concisely, as well as avoiding constructions that hamper caching by external proxies: all pages have independent, separate URLs and no articial distinctions between maps and individual topics are required.

For general navigation the user starts with a map and a listing of its contained topics and henceforth moves over to other involved topics or maps by following hyperlinks.

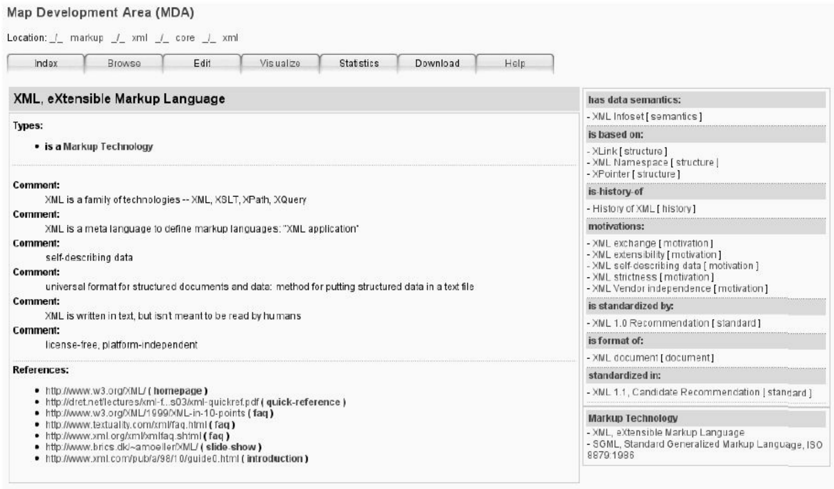

Fig. 1. Topic Map Browsing.

Figure 1 shows a typical topic display with the main pane displaying the subject, XML, and the surrounding panes to the right covering associations that involve the subject (in association displays the focus topic is suppressed if it occurs as a player).

In addition to this layout we also incorporated a visual display mode in which the topology of a topic's associations are presented as an SVG graphic. This experiment was only moderately successful, because while the topology view provides a fast way of following through associations, the state of SVG support in common browsers leaves a lot to be desired and presenting both topology as well as textual content proved to be very cumbersome.

Besides the exploration of pre-existing maps the MDA also covers creation and maintenance of maps, but for obvious reasons this functionality is restricted to registered users. Maps can be uploaded, converted and edited within the MDA, but editing of maps via the web is supported only in a limited fashion because we consider real text editors vastly superior to a web interface.

Because AsTMa= maps are plain text les, we enjoy the benets of being able to use a version control system (namely CVS) eciently to keep track of revisions to our maps and maintain an audit trail. On the other hand we need to avoid the performance impact of repetitive parsing of maps within the knowledge server applications and therefore the CVS-stored maps are not used directly. Instead the maps are pre-parsed and then stored as binary cache objects.

4 Creating Teaching Materials

Our preferred teaching style requires only limited graphical sophistication in lecture materials, and therefore coarsely represented knowledge from Topic Maps is very suitable as basis. The ease of including links to external materials adds to the appeal of using Topic Maps for this purpose, but there are some obvious downsides to this approach: map-based material tends to be primarily textual. We consider this to be an acceptable property because our lectures make heavy use of interactive demonstration and whiteboards.

4.1 Presentations

Our aim was to use our one knowledge store in as many contexts as feasible, notably among them preparation and management of lecture materials. For lectures we primarily use slide shows that outline the intended material very roughly, which are then expanded by interactive demonstrations and backed up by directly linked-in external materials. Graphical elements like diagrams are usually not the dominant aspect of the content but where necessary we embed externally created graphics (referenced in the maps as occurrences) in our materials. In this fashion Topic Maps provide us with sucient exibility to express what we want to present.

Generic, unsequenced map browsing as described before is not ideally suited for use as lecture material because of the lack of inherent sequence or organization among topics and the lack of tailoring towards specic audiences. Most people are familiar and comfortable with linear, conventional slide show formats and we agree with this preference. Our knowledge server was therefore expanded to cover the creation of such materials from topic maps as well.

To extract information from maps and arrange it meaningfully in a linear fashion, a sequencing mechanism needed to be developed. We also found it necessary to incorporate a ltering mechanism to narrow the (potentially extensive) material that a map contains down to a more appropriate degree for particular audiences. The more connections between topics, the higher the semantic value of the map { but the more disorienting unrestricted navigation of the topic itself becomes. For example, a map deeply covering XML technologies may also be used in subjects where only a very broad overview of XML is to be presented and where therefore the arcane details are likely not useful in providing *relevant* information.

Finally, we found it desirable to include a means of specifying some rendering information so as to control how particular aspects of a topic or association are shown on screen. This also includes the capability to include external graphics in the result material.

For the purpose of storing these lters and sequences separate from an actual map we have developed the concept of a view of a map[5], which contains a snapshot of the map together with relevant ltering, sequencing and formatting directives to transform the map into a form suitable for slide-show presentation. These views are self-contained and can be resynchronized with the respective map after updates, and the view information will be kept intact.

This abstraction layer was developed practically into TM::View[6], a Perl module for managing such views. In order to create a slide show from a map, the presenter creates a view of said map and then selects the appropriate elements to include in the view.

Different from the editing of maps which we expect to happen primarily outside of the knowledge server application, we decided to cover all necessary aspects of map views directly within the knowledge server. The reason for this is that unlike the authoring of topic maps themselves, the generation and editing of such views is a suciently visual task to benet from inclusion in the web interface.

The screenshot in figure 2 gives a typical example of editing actions on an individual topic in a view: aspects of the topic can be reordered, excluded (greyed out), shown bulleted or emphasized and so on. For the sequence of topics in a view only order and inclusion need to be edited. The right panel shows the resulting slide (as rendered by LaTeX Beamer).

Once a view is created and adjusted, it can be exported to a simple intermediate format named *listlet* which describes individual slides. This XML-based format allows us to separate the view system from the many different output mechanisms, and adding new output mechanisms is thus a straightforward exercise of providing an XSLT stylesheet transformation for the nal output format. Currently our knowledge server environment supports the output formats DocBook, LaTeX Beamer, AxPoint and HTML. HTML output is possible in two fashions: for interactive online viewing where a presenter can step through the slide show page-by-page, or with the whole slide show on one page in a format suitable for handouts.

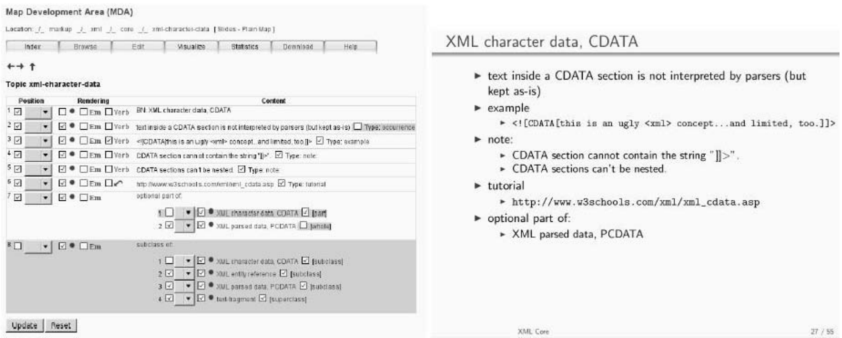

Fig. 2. View Editing Interface and resulting slide.

4.2 Background Reading Material

Besides presenting maps in form of ne-tuned and linearized slide shows we also made our maps publicly available and suggest them to students as sources of background material for home study. The highly hyper- linked presentation of topics and easy access to related aspects of concepts has proven to be quite useful as an encyclopedic resource.

As an extra resource for self-directed reading the unsorted nature of topic maps is also less of an issue, especially when used only as a secondary, in-depth augmentation of the normal slide show materials.

5 Topic Maps as Assessment Mechanism

We think that Topic Maps present some fairly unique advantages for the assessment of students' mastering the subject material, which led us to adopt the creation of topic maps as assessment format for some postgraduate and a few undergraduate subjects (excepting rst year courses).

Students are tasked with the research of some subject-related area followed by the creation of a topic map that covers their research; the map is the sole deliverable instead of the more common essay submission. The assignment topics are usually specied fairly broadly and students get some leeway regarding whether they want to cover the topic with an emphasis on broad and comprehensive or narrow and specic coverage.

Over the last six years we have had very good results with this approach. The most substantial benets we have experienced are related to infrastructure and knowledge reuse, enforced conciseness and precision, a low entry barrier and relatively high language independence.

5.1 Low Entry Barrier

It takes only very little time to get accustomed with Topic Maps as a knowledge store, because the author-visible framework is quite minimal: one needs to know the main dierences between topics and associations, a rough idea of the few syntactic rules that the particular Topic Map notation uses is also necessary and beyond that only a simple text editor is required.

In our experience it is sucient to provide students with about 1 to 2 hours worth of Topic Map indoctrination before they are suciently primed for independent operation. Usually we use examples of our own subject material maps for this introduction session. The overall advice on procedures is also very brief and easy to deal with: We suggest that students start with collecting relevant keywords, concepts or topics, followed by the addition of extra information for each item (\just like an annotated list of bookmarks"). After this step, one needs to identify the *relevant* and *important* topics, which students are then expected to bind to other topics using associations to represent the essential relationships and causal connections. These steps do not require computer interaction, and only the nal representation of the thusly-built \mind map" requires knowledge of a Topic Map notation's syntax and details.

The first topic collection step rarely poses any problems, and with diligent students usually yields a lot more material than can be included feasibly. The second step is the crucial part, because meaningful associations are the main carriers of knowledge in the Topic Map arena. In our experience lazy students usually only bother with a rough taxonomy or categorization to link topics in a thin fashion, because that requires least knowledge of the problem domain and minimal eort: stating that something is a part of something else, or belongs to a particular class is trivially simple.

Diligent students, on the other hand, start digging deeper at that point and usually come up with lots of interesting connections. With a bit of experience it becomes easier to single out the most pertinent of these connections, which then become candidates for explicit representation in the nal map. This selection phase is the most crucial and also the one where most (minor) problems arise: While the notion is quite natural that a topic in an association must have a role that it plays, it is not trivial to come up with good, descriptive names for those roles (except maybe in the case of \natural" relationships like author/work or parent/child). Facing the need of nding good association and role names causes most support requests by students, but generally this hurdle is easily overcome with a bit of exercise.

The final step, actually writing the map, is the least of the students' worries in our experience. The simple formats and the few syntactic restrictions imposed by the shorthand notations win greatly over the extremely verbose standard XML-based formats, and having access to our knowledge server and general parser software helps students in getting the last syntactic bugs out quickly.

5.2 Enforced Precision

Creating a high-quality topic map requires that the author has acquired a substantial degree of knowledge of the subject, and also understood and identied the important relationships between concepts. Expressing this knowledge as a map results in a very concise representation, which we believe is a major benet of Topic Maps for assessment: it is quite hard to inflate a map's content articially as is often done in essays. (Or conversely, it is relatively easy to spot badly-connected, irrelevant parts in a map.)

Furthermore, knowledge represented in a map captures precise, individual facts or connections. This makes assessing a map author's understanding of the subject easier, because whether the author has identied and included the salient subject features in the map is very evident - much more so than when hidden somewhere in an essay.

To promote precision and relevance in maps, our guidelines to students usually include the recommendation that they focus on expressing important facts using associations (instead of attaching long texts to individual topics) and that they use a variety of dierent associations (instead of relying solely on classication and inclusion relationships).

5.3 Reuse

Requiring students to work on Topic Maps allows us to provide them with our considerable infrastructure for map storage, display and querying and, of course, syntax-parsing as well. This makes it easier for them to work with their maps during the nal stages of their assignment, and also gives them ample examples of our best current practices as evident in our public maps.

Interestingly enough the publicly available maps seem not to tempt students into plagiarism too much; we believe this is strongly related to the conciseness and specicity of the format.

Furthermore there is a potential for knowledge reuse going the other direction, too: among the top few percent of our students we occasionally receive material that is of sucient quality to be used as base for our own lecture materials. While we rarely get to reuse a complete student map, the Topic Map model nevertheless makes it feasible and easy to reuse partial maps in dierent contexts.

5.4 Language Independence

Our academic environment includes a substantial percentage of exchange students whose English skills vary greatly. We feel that excelling at essay-type assignments is substantially harder for non-native speakers. On the other hand the Topic Maps' concise nature of representation requires less highly developed language skills and thus levels the playing eld some. In the area that tests an author's ingenuity most (naming of associations and roles), in our experience there is little dierence between the maps created by native and non-native speakers.

6 Summary and Future Work

Over the last six years we have developed and rened our knowledge server environment to the point where it is suciently exible and robust to be used not just be the authors but also by students who are generally unexperienced with knowledge management mechanisms. Besides serving as a knowledge store that can be browsed directly at leisure, the server environment also evolved into a means of eciently creating simple but appealing slide show presentations from knowledge represented in the store. This semantically-enhanced representation has shown substantial benets and advantages regarding the long-term maintenance and upkeep of teaching materials, especially where evolving technologies are to be covered.

Additionally we have had positive experiences with the creation of Topic Maps as an assessment format, because the concise, straight-forward nature of the Topic Map model tends to force students towards examining and identifying the essential aspects of whatever they are to cover.

Our plans at this point include mainly minor improvements like better support for embedding external material in slide shows, as well as conceptual adjustments in the way maps are arranged and cross-linked on the knowledge server.

References

1. Barta, R., Heuer, L.: AsTMa= 2.0 language denition. Bond University (2005)
2. Barta, R., Zangerl, A.: Evolution of a Perl-based Knowledge Portal. In: Proceedings of the Open Source Developers Conference. OSDC 2004. (2004)
3. Barta, R.: Topic Map modules for Perl. http://search.cpan.org/~drrho/TM-1.44/ (2008)
4. Ahmed, K.: Topic Maps 4 Java. http://tm4j.org/ (2001)
5. Barta, R., Zangerl, A.: Slide Shows Meet Topic Maps. In Arabnia, H.R., ed.: Proceedings of the International Conference on Information and Knowledge Engineering. IKE'03, June 23 - 26, 2003, Las Vegas, Nevada, USA, Volume 1, CSREA Press (2003) 379{383
6. Zangerl, A.: Topic Map View module for Perl. http://search.cpan.org/~alphazulu/TM-View-0.3/ (2008)