Bookmap – A Topic Map Based Web Application for Organising Bookmarks

Tobias Hofmann and Martin Pradella

Bauhaus University Weimar, Faculty of Media, Bauhausstr. 11, 99423 Weimar, Germany
{tobias.hofmann@medien.uni-weimar.de
martin.pradella@com2-gmbh.de}
http://www.uni-weimar.de/medien/cg

Abstract. This paper proposes a basic Ontology for use in Topic Maps storing semantic information on bookmark collections. Furthermore, we introduce a data model allowing to implement such a system on a LAMP (Linux, Apache, MySQL, PHP) platform, extended with the Cake-PHP framework. A prototype has been developed as proof of concept, where the use of AJAX and drag and drop capabilities in the browser resulted in a good user experience during a preliminary user evaluation.

1 Introduction

As part of the work for a diploma thesis, we needed to have a bookmark organising system which is server-based and accessible from everywhere. In this system, we wanted to be able to have a hierarchical representation, like we know it from current browser and bookmark systems: trees with folders, subfolders, and bookmark items. Such bookmarks should be imported from existing collections of bookmarks, but also be added at a later date, in a very simple manner, so the system can be used by inexperienced users.

Furthermore, we wanted to be able to connect folders and single items using semantic associations. This is why we decided to include Topic Maps into the system design, which ended up as a web application, due to the internet nature of bookmarks. As a result, all user interaction was to be done whithin a browser window, accessing the server from everywhere. Finally, bookmarks were to be treated differently, depending on their privacy status, and possibly reused by other users of the same system. The name of this system was decided to be bookmap.

In this paper, we will now describe the requirements, some of the design decisions, and architectural elements of the bookmap software.

2 Previous and Related Work

Passin describes in [1] a topic maps based system to organise bookmarks, and implements such a system using the TM4Jscript library [2]. Here, the data model focuses on the representation of hierarchies, and integrates Passin's own concept of 'And-Terms' to allow for cross references. As we did not intend to use the hierarchic tree as a central element, but saw it more of a concession to users' current understanding of classifying, Passin's model did not come to use.

Then we have, of course, social bookmarking portals, enabling users to have a central bookmarking organisation. Del.icio.us as one of the first and surely coining representative of its kind has, as Matt Biddulph states [3], three axes: User, Tag, and URI. The users make up the social element, building an information sharing community. The Tags allow the users to classify information differently from, say, taxonomies, in a seemingly more convenient and surely more popular manner. Finally, the URI as the central element of organisation comes in.

Recently, a site with the name Fuzzzy.com has come to life, which came to our attention when development work on the current state of the bookmap system had been finished. Fuzzzy seems to build an ontology by using tags as topics, and by giving the user the opportunity to create horizontal and vertical associations between those tags. Later, the bookmarks are tagged and thus attached to the ontology.

This is different from our approach: We have a bookmark, which is represented by a topic, a basename and the topic's subject identity (URI). The basename is the topic's and thus the bookmark's name. The subject identity of this topic is the URI. Other users may have their own bookmark, of the same URI, but with another basename. This allows a very individual organisation of private or partly private bookmarks. Furthermore, Fuzzzy seems to suggest to the user to work without a hierarchical representation many users have come to rely on.

3 Motivation and Goals

One of the objectives of the diploma thesis was to identify a scenario in which Topic Maps would be put to good use. The developed software should itself be and rely on open source software. Eventually, a prototype was to be built and evaluated, time permitting, against the requirements collected around a user centered approach.

Technically, and after quite some time, it was decided to go with a LAMP based backend, also due to the high degree of availability in the hosting market. On top of this, and to provide for a flexible architecture with clear structures, we would use the MVC (Model-View-Controller) paradigma, to separate the three layers storage, system logic and presentation, by using the CakePHP framework [4]. The data and logic provided by the backend was to be presented to the user in a browser window, on as many OS platforms as possible, and closely adhering to W3C recommendations, which led us to Mozilla/Firefox as the browser of choice. Eventually, we would want to have a good desktop integration and high ease of use by permitting the user to drag and drop items, also from the desktop to the browser window, in our aim to meet the user's expectations.

4 Ontology

We want to represent bookmarks in a topic map. To do so, and as mentioned above, we use for each user and each URI a topic of type item which we call bookmark. This bookmark has a basename, and a subject identifier with the bookmark's URI (see (Fig. 1)).

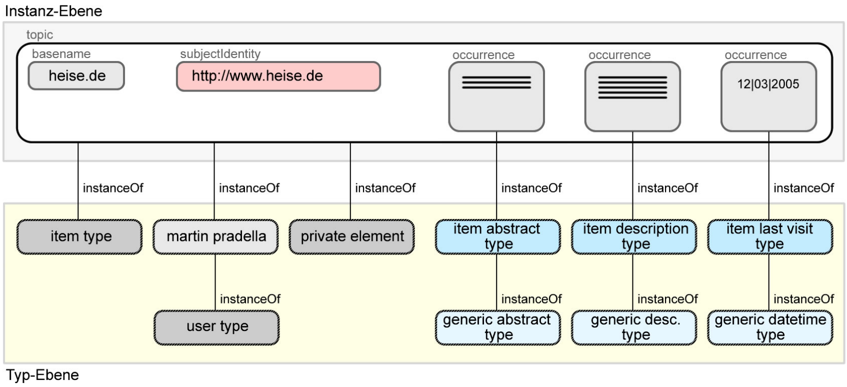


Fig. 1. Visualisation of the data structure of an item-topic

Such bookmarks or items can be connected using semantic associations between two bookmarks. The ontology provides a set of defined association types (see (Table 1)) including role types, which we found to be sufficient to express the vast majority of relations our users wanted to express. You will find two types of associations, directed, with different role types, and inderected, with only one role type. Furthermore, associations in bookmap are always 1:1, never 1:n.

5 Database Model

The design of the database schema is of great importance for the prototype, as its mapping of the Topic Map Model has to support the structural flexibility of the Topic Map Paradigma to the full extent. This is why following the proposition of Mugnaini [5] we extended his database scheme to our needs. This allows us to map the complete Topic Map Model, and represent each Topic Map construct in the relational database. The result of this work can be seen in Fig. 2.

6 System Architecture

Fig. 3 shows the technology structure, fitted for the structural parts of the web application.

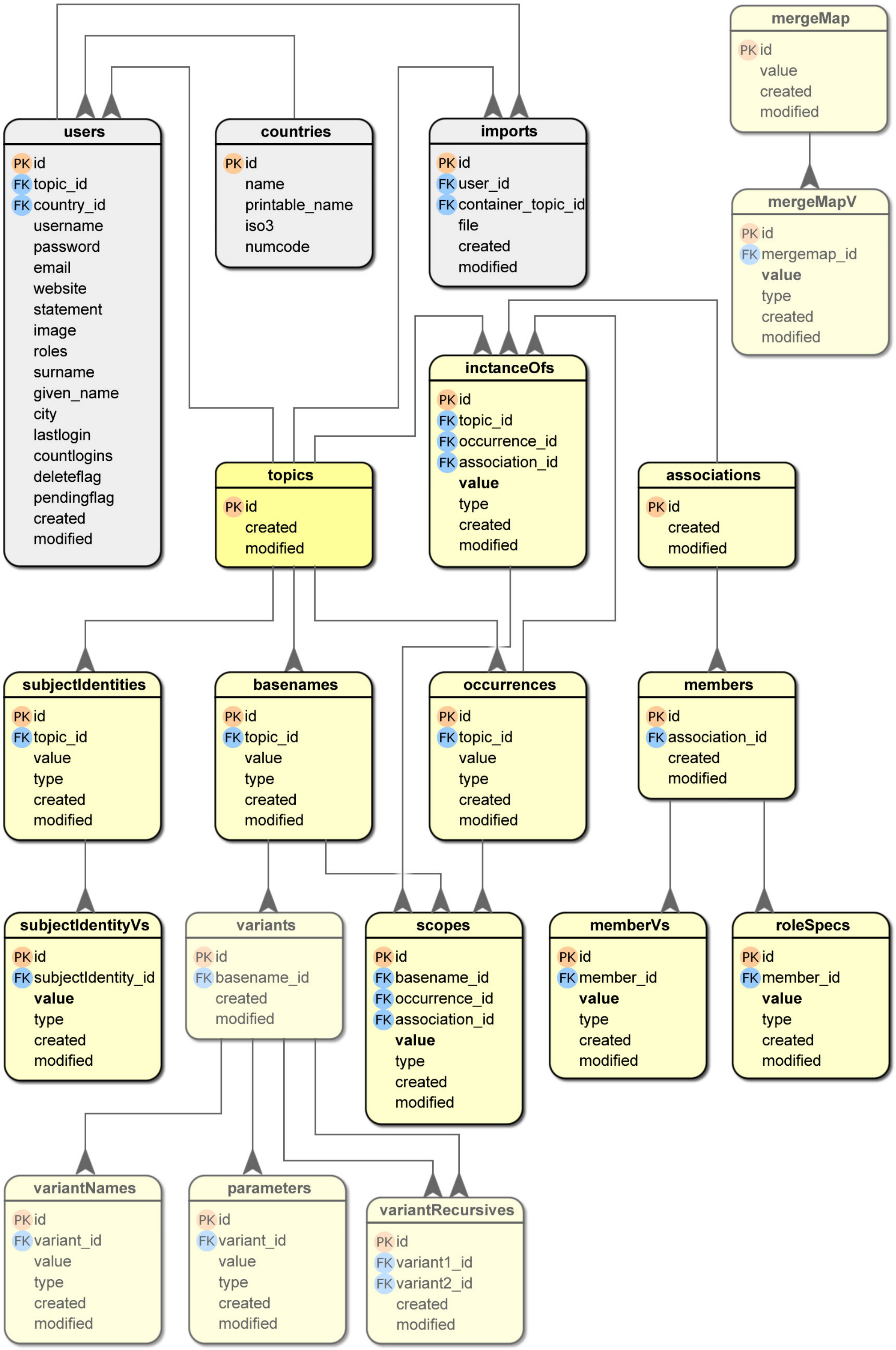


Fig. 2. Visualisation of the database structure

Table 1. Associations provided by the Ontology, including role types

AssociationType	*Roletype A*	*Roletype B*
explanation	explains	is explained by
description	describes	is described by
interpretation	interpretes	is interpreted by
demonstration	demonstrates	is demonstrated by
discussion	discusses	is discussed by
recommendation	recommends	is recommended by
summary	summarizes	is summarized by
sequence	precedes	to follows to
part-whole-relationship	consists of	is part of
weblink	links	is linked by
tagging	tags	is tagged by
confirmation	confirms	confirms
contradiction	contradicts	contradicts
relation	is related to	is related to
equality	is equal to	is equal to

We use a standard current LAMP platform, extended by the use of the CakePHP, which allows us to have structured and enhancable code in PHP. Basic classes of the Cake framework have been used to derive, amongst others, helper classes, which allow the implementation of AJAX functionality. An example of how such helper classes come to use can be seen in Fig. 4.

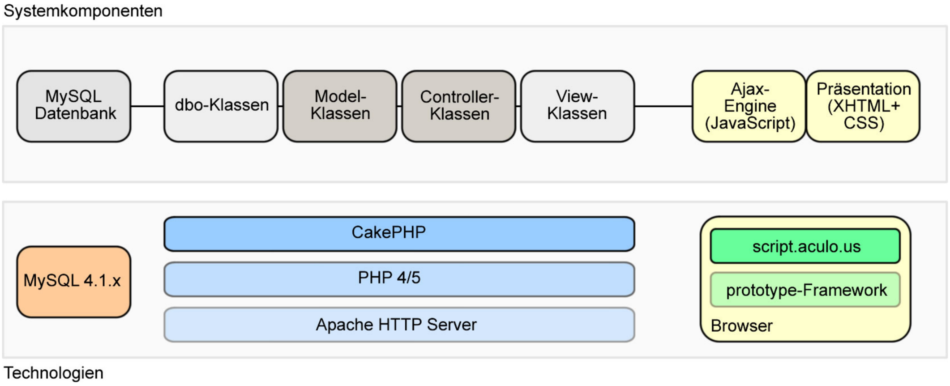


Fig. 3. Visualisation of the system architecture

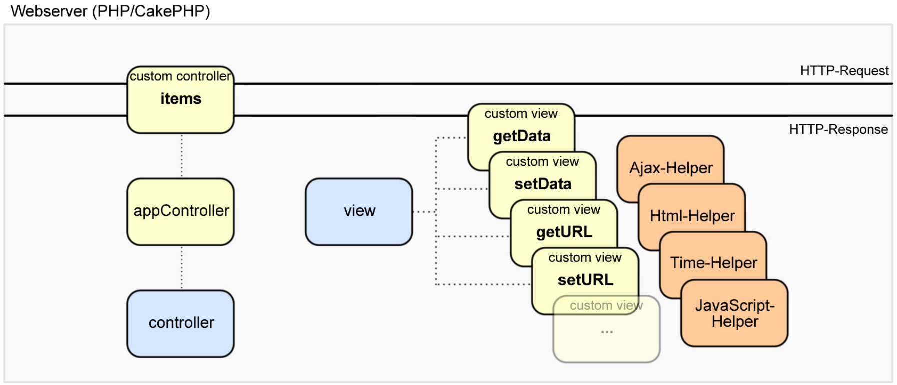


Fig. 4. An example of the interoperation of controller, view, and helper classes.

7 Discussion

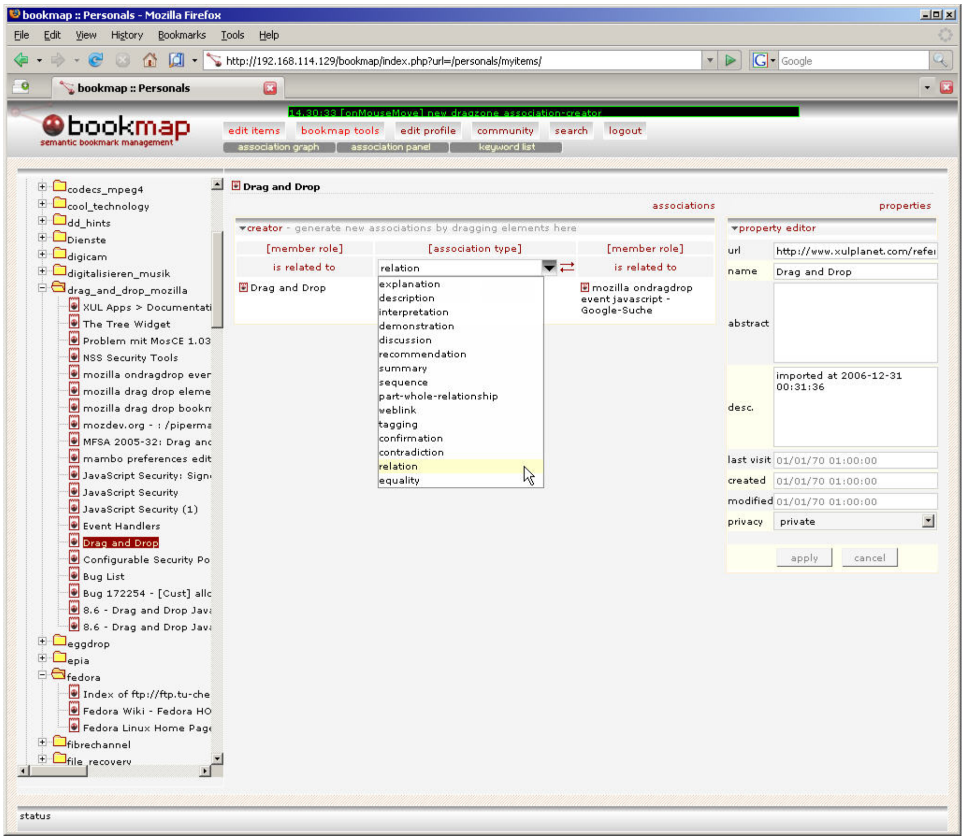
When the decision on the underlying technologies had to be made, we focused on a lightweight and simple solution, which meant to avoid Java, and (unfortunately) TM4J [6] as a Topic Map API with it. This meant that we had to develop our own solution for the backend, which, of course, was met with the arrival of the TMAPI for PHP [7], once we were almost finished. Today, we would naturally recommend using available solutions, which would also save lots of time (and nerves).

Next, we found that there is still room for improvement concerning data storage. We use large amounts of memory and processing ressources to be compatible to most of the XTM syntax. While this is very convenient for a prototype, as it keeps you flexible, it is also quite expensive when it comes to complex database queries and the added time resulting from the complexity. Once the extension of the prototype's functionality has calmed down, it is recommended to optimise the data model.

The decision for an AJAX frontend is something we are still happy with, as it allowed us to implement platform independent and sophisticated user interfaces, thus mostly minimising shortcomings of classic web applications. Those interfaces include drag and drop, context menues, and keyboard shortcuts for actions like renaming. It must be said, though, that this benefit comes with the price of essential development cost, which, especially for a project of such limited resources, has to be taken into account.

Fig. 5 shows the prototype as a user would see it in a firefox window. Currently, the user has to login onto the platform using a password, and multiuser capabilities are present in the data modeling, but sharing bookmarks with other users is not yet implemented. To the left, we see the tree, as known from other applications. In the middle, we create associations between items by dragging them from the tree onto both sides of an association type, which we then choose from a dropdown list. Finally, we determine which item plays which role, and create the association. Properties of the item can be edited on the right hand side. Plans to also include, possibly as a plugin, a graphical visualisation of parts of the underlying Topic Map had to be postponed, as had to be a linear history of bookmap items already visited.

An informal user evaluation has shown that users quickly grasp the elements and functionality of the user interface, and quickly achieve usual interaction goals (Search, Sort, Move, Follow Assoc., easily. The system has been tested to import up to over 6000 items, which it manages to do in under 30 minutes (VMWare virtual Linux server, running on Dell Latitude D610, 1,73GHz Pentium Monocore Processor), and then allows the user to interact in a fluent manner.



http://192.168.114.129/bookmap/index.php?url=/personals/myitems/

14.30:33 [onMouseMove] new dragzone association-creator

Drag and Drop

[member role]	[association type]	[member role]
is related to	relation	is related to
Drag and Drop		mozilla ondragdrop event javascript - Google-Suche

url	http://www.xulplanet.com/refe
name	Drag and Drop
abstract	
desc.	imported at 2006-12-31 00:31:36
last visit	01/01/70 01:00:00
created	01/01/70 01:00:00
modified	01/01/70 01:00:00
privacy	private

Fig. 5. A screenshot of the final application.

7.1 Conclusion and Outlook

We have presented the bookmap system, a web application to import, organise and present website bookmarks in both, a hierarchical and semantically associated manner. The system is a prototype and implemented using an open source LAMP environment, extensions implemented using the CakePHP framework, PHP- and javascript. The resulting user interface is available via the Firefox web browser and delivers a rich user experience including drag and drop functionality via AJAX.

Currently, it is planned to publish the current source code on sourceforge.net within the next two months. After that, and depending on the feedback and input of the open source community as well as the topic map community, the following topics could be next steps for bookmap: Implementation of the already prepared multi-user features; evaluation of TMAPI-PHP, and possible redesign using TMAPI-PHP; Specification, design and implementation of a visualisation plugin; Specification, design and implementation of a Firefox extension/addon to enhance and improve the desktop integration capabilities.

This is why the authors are happy to hear from the readers, and look forward to possible cooperations to carry on with the development of what we see as a promising system.

References

1. Passin, Thomas B., 2003: Browser bookmark management with Topic Maps. In Extreme Markup Languages. URL http://www.mulberrytech.com/Extreme/Proceedings/html/2003/Passin01/EML2003Passin01.htm [Last accessed, like all other online references, on 22. 06. 2007]
2. Johannesen, Alexander und Thomas B. Passin, 2003: TM4Jscript Topic Maps For JavaScript. Techn. Rep., Open Source Technology Group - Sourceforge.net. URL http://sourceforge.net/projects/tm4jscript
3. Biddulph, Matt, November 2004: Introducing del.icio.us. Techn. Rep., OReilly Media, Inc. - XML.com. URL http://www.xml.com/pub/a/2004/11/10/delicious.html
4. Woodworth, Garrett J., Larry E. Masters und Matti Putkonen, 2005: CakePHP - The Rapid Development Framework. Techn. Rep., Cake Software Foundation, Inc., Nevada. URL http://cakephp.org/
5. Mugnaini, Luca, 2003: Mapping Topic Maps on Relational Databases. Techn. Rep. URL http://www.geocities.com/xtopicmaps/mapping_xtm_on_databases.html
6. Fröhlich, Christoph und Kal Ahmed, 2004: TM4J Topic Maps For Java. Techn. Rep., The TM4J Project Group. URL http://tm4j.org/
7. Holje, Espen und Johannes Schmidt, 2005: TMAPI/PHP. Techn. Rep., Open Source Technology Group. URL http://sourceforge.net/projects/phptmapi/