TMNet - Distributed Viewing and Editing of Topic Maps in the World Wide Web Environment

Sebastian J.F. Fudickar[1], Klaus Rebensburg[2]
[1] Universität Potsdam / Institut für Informatik, Potsdam, Germany
[2] Technische Universität Berlin / FSP-PV Research Center, Berlin, Germany

***Abstract* - Since the Topic Map standard describes a prospective knowledge-structuring model that can be used in a huge variety of knowledge domains the amount of applications utilizing this standard grew enormously. Anyhow, as far as we know, there is no distributed editor of Topic Maps available on the market, that supports an intuitive possibility to manipulate or visualize Topic Maps in the World Wide Web environment. Therefore, TMNet intends to close this gap. As a result, a high performing distributed editor is achieved, with which web-based e-learning environments as well as knowledge environments can be enhanced by supporting semantic data descriptions of their knowledge structure. Additionally TMNet enables the user to extend the structures in an easy way and gives the web developer an opportunity to integrate it into websites easily.**

***Index Terms*— distributed, editing, semantic network, Topic Maps, World Wide Web**

I. Introduction

The Topic Map Standard [1] describes a prospective knowledge-structuring model in the world of semantic networks but is not limited to this domain. It is also applicable in domains like information indexing, thesauruses, glossaries, for the generation of classification-based ontologies and others.

By the usage of Topic Maps, a variety of fields of knowledge representation can be interconnected. The resulting inter-domain knowledge base may contain an unlimited amount of knowledge and the structure can include a high grade of complexity. This is extended by the possibility of merging different instances of Topic Maps, which represents an additional possibility to increase the amount of included knowledge.

The supported linkage to external information-resources specified by a Topic Map's implicit structure that resides outside the information resource enables the possibility to link from the resulting knowledge structures to external information resources, including media (like images or videos), and can therefore be used to give a semantic meaning to any media.

By this, Topic Maps seem to be well designed for most fields of knowledge representations. These advantages, like the usage in manifold domains, the supported high complexity, the linkage to external media and high usability of this standard, lead to a significant amount of programs that utilize the Topic Map standard as a knowledge structuring model. These tools include Topic Map editors, semantic desktops, library management tools, social networking tools, and far more.

The Topic Map editors we found on the market[1] usually are implemented as stand-alone applications. This approach seams not to be optimal in the knowledge domain, because it complicates the generation of Topic-Map based knowledge structures in groups or communities. However, a distributed approach that supports the usage in the World Wide Web environment seams to be a much more encouraging approach for this scenario.

We are going to introduce our application TMNet that enables the manipulation of Topic-Map based knowledge structures by groups, technically by using a distributed approach. Therefore, major criteria are introduced and our approach of the solution is shown. The subchapters introduce a variety of aspects of TMNet en detail and an outlook is given.

II. TMNet

Applications that focus on the generation of structured data should support a group or community-based usage, to increase the quantity and quality of the contained information and the resulting knowledge structures. Therefore, TMNet attempts the community-based generation of semantic knowledge structures by using a distributed approach that supports the usage in the World Wide Web environment. Additionally TMNet supports concurrent working on the knowledge structures, without demanding the concentration of the user for this task.

TMNet utilizes the data storage format that is standardized as *XML Topic Map* (XTM) that enables an XML-based storage of Topic Maps in files. Because of an increase of the data access performance, a database storage of Topic Maps seams to be reasonable in this application. Since this concept of realization is not yet standardized, we enabled an easy integration by defining appropriate interfaces, which can be used later on. Thereby it must be considered, that the usage of this database approach may result in a more complex integration of TMNet into websites, since the database must be installed and configured additionally.

Short data transfer delays are important criteria for the usability of distributed applications. Since Topic-Map based knowledge structures can reach a huge quantity and through this also huge storage-size, TMNet includes

[1] Like *Two More* (http://www.pi.informatik.tu-darmstadt.de/se2004/byteme/)

caching functionality in order to reduce the necessary data traffic between the components as well as loading times. Additionally an internal data model that benefits in a reduced data size is used for communication between the components. Therefore, the information structures are reduced to the essential elements, which are additionally represented by an object-based data structure instead of an XML-based one.

For knowledge-representation scenarios, we want to present the information in a more appropriate way in order to enhance the human knowledge-generation process. Quillian et. al. [2] proved that the optimized way for a human to acquire object knowledge is by visualizing relations to other objects. Therefore, in TMNet a graph-based approach for the visualization of the relations of the objects is used.

Furthermore the study of Rispen et. al. [3] explained that the recognition of objects by navigation through knowledge representations, which are visualized as three-dimensional graph structures, usually is more effective than in two-dimensional graph- or list structures. Therefore, TMNet supports a pseudo three-dimensional visualization engine with different layouts next to list views and a search-functionality. By this approach, the user may navigate through the presented knowledge structures in various ways.

In order to reduce the complexity of the editing process the user interface of the Topic Map editing functionality is embedded into a wizard structure. The possibility of editing a Topic Map is depending on the preferences setup during the integration of the TMNet Applet component into a website.

The integration of TMNet into HTML-based websites is straightforward and enables any web developer to utilize it. By utilizing common standardized technologies like Java-Applet and Servlet technologies, the onetime installation by the web developer just needs a Servlet supporting Server like Tomcat.

In TMNet these features seam to be a much more encouraging approach than a standalone application with similar functionality. Opposite to other applications the user of TMNet is able to edit Topic Maps 'on the fly' without a previous local installation process. By short response times during editing and an intuitive visualization as well as controls, the users´ attention may mainly focus onto the represented knowledge.

The following subchapters show aspects of the TMNet application in more detail.

A. *Distributed Characteristics*

TMNet is based on Servlet- and Applet technology, which enables the integration of the characteristics mentioned above. As a result, a distributed editor for Topic Maps is achieved, with which web-based e-learning environments as well as knowledge environments like Wikipedia can be enhanced by supporting semantic data descriptions of their knowledge structures.

The TMNet architecture can be basically divided into two components, as it is shown in Figure 1:

- The server-sided component is based on the Java Servlet technology and includes the Topic Map managing functionality, including conversion of Topic-Map based knowledge structures into the internal format, the authentication management, the version management of the knowledge structures,

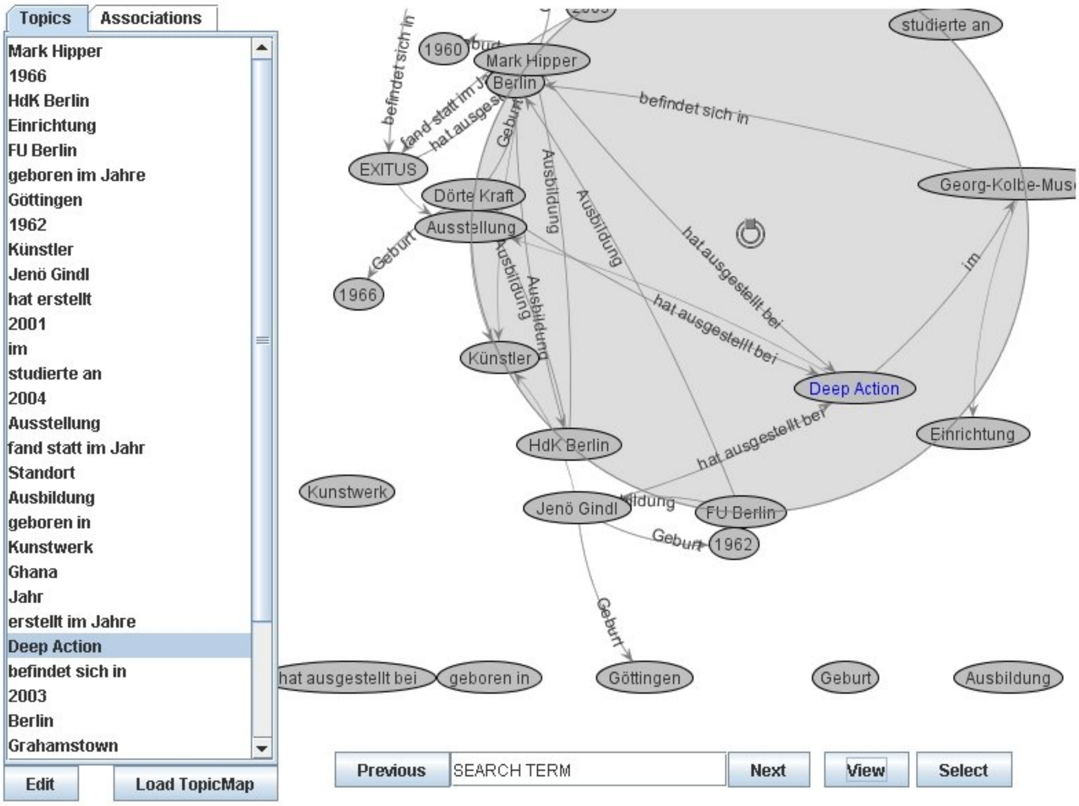

Figure 1 The TMNet GUI for the navigation through Topic Map based knowledge structures

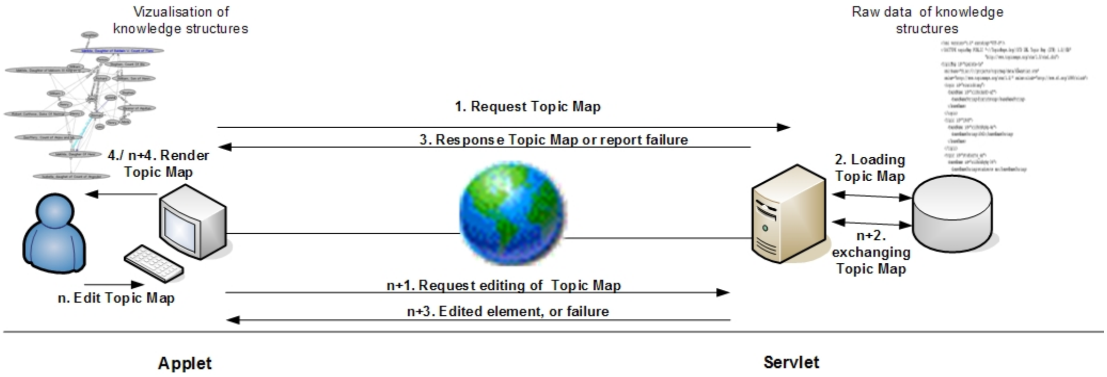
Vizualisation of knowledge structures
Raw data of knowledge structures
1. Request Topic Map
3. Response Topic Map or report failure
4./ n+4. Render Topic Map
2. Loading Topic Map
n+2. exchanging Topic Map
n. Edit Topic Map
n+1. Request editing of Topic Map
n+3. Edited element, or failure
Applet
Servlet

Figure 2. The concentrated communication workflow of TMNet

pre caching of knowledge structures as well as the loading, extending and saving of the Extended Topic Map (XTM) based files.

- The client-sided component is based on Java Applet technology and provides the user interfaces for the editing of Topic Maps. Additionally it contains visualization functionality of the knowledge structure and caching. By limiting the delay of response to a minimum by reducing the communicated data between both components to a minimum, the caching functionality increases the usability of the application significantly (Chapter C).

The communication workflow between both components of TMNet can be concentrated to loading and editing of a Topic Map based knowledge structure, as it is introduced in Figure 2. Additional cases like user authentication or the update mechanism are minor important, since their usage occurs less often.

When a Website including the TMNet Applet is opened, the Applet is loaded, and requests a representation of a Topic Map that has been predefined during the integration process of the web designer, from the TMNet Servlet. The TMNet Servlet checks if the requested Topic Map has already been loaded and cached. The requested Topic Map is then responded to the Applet, in which the cached Topic Map is utilized preferably. Then the TMNet Applet renders the data and displays it in various ways, like introduced in chapter B.

Another important case of communication between the two components occurs during the editing process. Thereby our approach reduces the amount of data that has to be sent to a minimum and gains thus optimal delays that are almost independent from the size of the Topic Map. Although TMNet supports different types of editing, like appending and removing of topics or associations, from the communication view all are treated in the same way. Different editing types can be distinguished through a flag that is handed over as well. After the user edited the Topic Map manually through the graphical user interface of the TMNet Applet a corresponding request is handed over to the TMNet Servlet component. This request includes the elements taking part in the manipulation and the flag that indicates the type of editing.

After the appropriate Topic Map is edited and saved by the TMNet Servlet component, another flag indicating the success of the editing process extends the variables of the request container. The extended variables then are responded to the instance of the TMNet Applet. If the editing process succeeded on server-side, the Topic Map is also customized and re-rendered on the Applet side.

The support of concurrency aspects has been integrated by a knowledge-data update mechanism. The update mechanism takes place before the editing process starts, since current changes in the distributed structure of a Topic Map are less important for the pure viewing. When a user starts editing, a request is sent to the Servlet in order to ensure that the local Topic Map still corresponds with the original one. The up-to-dateness is checked by an ID, which is incremented at both sites during the editing process. If this ID does not correspond between the requesting applet and the responding Servlet the Topic Map based knowledge structures must be refreshed on the specific applet by a re-transfer.

B. Visualization Characteristics

In order to enhance the knowledge-generation process in e-learning and knowledge-representation scenarios we want to present information in a more appropriate way that enables the user to acquire the represented knowledge as easy as possible. One of our approaches reduces the attention needed by the user for the user interfaces in order to enable more attention to the presented information itself. The success of this concept is shown in the past for instance by the DeepaMehta [4] project.

Therefore, TMNet visualizes information in a graph-based way – representing semantic structures. In this graph-based presentation, the user navigates through the knowledge structures and experiences by this the relations between objects ‘on the fly’. The user may switch between a flat two-dimensional and a pseudo three-dimensional visualization that is based on a fisheye algorithm (see Figure 3). Additional information can be experienced by selecting the objects in the graph and by presenting it separately.

The information objects of the knowledge structure can be reached in variable ways. Next to the graph-based approach, it is also possible to selected specific objects through a list or integrated search functionality.

The look-and-feel of the front-end of TMNet is also quite easily changeable by the Web designer through parameter settings.

TMNet does not necessarily support the editing functionality. This functionality can be switched “on” or “off” by the web designer, who integrates the TMNet Applet into his Webpage. If the editing function is not

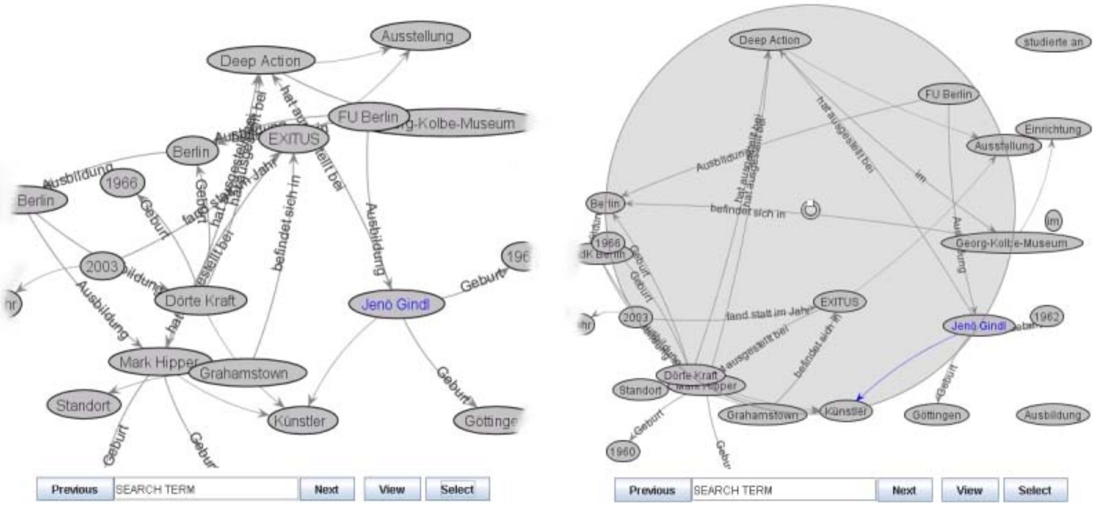

Figure 3. The different graph-based visualizations of TMNet - left: two dimensional, right: fisheye

supported, the Applet can be used for pure viewing of Topic-Map based knowledge structures.

The provided editing functionality of the knowledge structures is embedded into a wizard-structure in order to enable the user to pass through this task in a structure way. In our view this approach reduces also the necessary grade of experience the user needs.

Additionally the provider of a Topic Map may integrate an authentication before the editing is possible. In this case, the authentication dialog appears when trying to edit the Topic Map based knowledge structure.

C. *Optimization Characteristics*

The architecture of TMNet is designed under consideration of the fact that Topic-Map structures may consist of a large amount of data (sometimes exceeding 20MB). Therefore, data is cached on server-side, so that a Topic Map has to be loaded only once from file in the lifecycle of a Servlet.

Additionally, Topic Map data usually are transferred once in the lifecycle of an applet. This is achieved by separately extending the knowledge structure in the Applet and the Servlet. This reduces the response time significantly.

In case of editing activities, only the information of the data-structure extension is sent back to the Servlet. Additionally the amount of data of the transferred data model is reduced in comparison to the loaded Topic Map model by the use of a non XML-based object model instead. This approach reduces the amount of data as well as the response time, which leads to a significant improvement of usability.

III. Summary

TMNet is a distributed application that enables the distributed manipulation and viewing of Topic Map based knowledge structures in groups and communities. The main criteria of such an application handling are minimum delays during data transfer, as well as intuitive user interfaces, an appropriate knowledge presentation and concurrency tasks. Considering these criteria, TMNet supports short response times during the editing process, a user-friendly user interface as well as a knowledge representation that is optimized for fast-learning achievements.

Another aspect of TMNet is the quick-and-easy integration of the application into a website including a customization to the actual requirements. Therefore, TMNet is based on Java Applets and Servlet technology, which fits these goals. The customization is supported through parameters that are integrated into the website.

As a result, TMNet is a contribution closing successfully the gap for communication of semantic structures between humans and computers. Because of semantic information added, future semantic-aware search engines could be able to deliver more precise results of e-learning web sites to their users.

IV. Outlook

In the near future, this application will be embedded into an art-network portal, where artists will be able to extend the knowledge base with their profession-specific biographical data. Thereby the application will be tested and revised in a real knowledge-based environment. Measurements and the user feedback will give tasks for additional optimization. The integration of additional functionality like the support of linking to external information resources will also be provided.

TMNet can extend the functionality of community websites dealing with knowledge like Wikipedia[2] with the feature of adding a computer and human readable semantic meaning to its content. The resulting benefit in form of semantic content descriptions will help community sites to be well prepared for the emerging semantic web and its search engines. Therefore, in our view TMNet is seen as an important step for future Web 2.0 websites.

[2] http://www.Wikipedia.org

REFERENCES

[1] International Organization for Standardization: ”ISO/IEC 13250, Information technology - SGML Applications – Topic Maps”, ISO, Geneva 2000

[2] Quilian, M.R.: Semantic Memory. In: M. Minski (ed.). Semantic Information Processing. MIT Press, Cambridge, MA (1968)

[3] Risden K., Czerwinski M. P., Munzner T., Cook D. B.: ”An Initial Examination of Ease of Use for 2D and 3D Information Visualizations of Web Content” Microsoft Research, Redmond

[4] Jörg Richter J. Völkel M., Haller H.: DeepaMehta -A Semantic Desktop, in Proceedings of the 1st Workshop on The Semantic Desktop. 4th International Semantic Web Conference (Galway, Ireland), volume 175. CEUR-WS, November 2005.

AUTHORS

Sebastian J. F. Fudickar is Master Student of the Computer Science Department of the University of Potsdam. He received his Bachelor Degree in Softwaresystem Engineering at Hasso Plattner Institute at University of Potsdam. Current research fields are: multimedia communication in MANETS, Web 2.0 and Semantic Networks.

(e-mail: Sebastian.Fudickar@gmx.de)

Klaus Rebensburg received a diploma degree in electrical engineering and the Dr.-Ing. (Ph.D.) degree in software engineering from Technische Universität Berlin, Germany in 1982. He is Head of FSP-PV Research Center/tubIT at Technische Universität Berlin. He is also teaching Computer Science as a Honorary Professor at University of Potsdam. His research fields are network technologies, multimedia applications and semantic media. (e-mail: Klaus.Rebensburg@tu-berlin.de)

Manuscript received 16 February 2007.