

Adding Meta Data to Documents for E-Learning: A Tool Evaluation

doi: 10.52987/edc.2025.003

Harald Wahl

University of Applied Sciences Technikum Wien
Hochstaedtplatz 6, 1200 Vienna, Austria
wahl@technikum-wien.at

Alexander Mense

University of Applied Sciences Technikum Wien
Hochstaedtplatz 6, 1200 Vienna, Austria

Martin Niederwieser

University of Applied Sciences Technikum Wien
Hochstaedtplatz 6, 1200 Vienna, Austria

Sonja Vollnhofer

University of Applied Sciences Technikum Wien
Hochstaedtplatz 6, 1200 Vienna, Austria

Shveta Sohal

University of Applied Sciences Technikum Wien
Hochstaedtplatz 6, 1200 Vienna, Austria

Sebastian Schwarz Mense

University of Applied Sciences Technikum Wien
Hochstaedtplatz 6, 1200 Vienna, Austria

ABSTRACT

Modern Web 3.0 learning environments require a solution for making all related documents, which may be stored and accessible through various systems, findable in a semantic way. This paper deals with the requirements of such a solution, which are based on a project from the University of Applied Sciences Technikum Wien and evaluates which of the existing software solutions on the market meets the needs, while only ready-to-use software and components are considered. Features and functionalities of four software solutions (Microsoft SharePoint, W3C Annotea, Ontopia Knowledge Suite and Caringo CASTor) are compared with the requirements. Possible usage scenarios provide hints and general thoughts for putting the project's concept into practice. An evaluation of all the software solutions considered shows clearly the advantages and disadvantages of them. It turns out that no considered software solution meets the needs perfectly, but this paper illustrates the best usage for e-learning purposes.

KEYWORDS: E-learning, Topic maps, Annotation, Semantic technologies

11 INTRODUCTION

Considering the development of e-learning platforms in universities or similar institutes and the need for a knowledge-driven solution for browsing e-learning content stored in various systems, a modern Web 3.0 learning environment is required. The University of Applied Sciences (UAS) Technikum Wien tries to find an approach to meet the needs of such an environment using semantic technologies (project “eLearning 3.0”). One important part of this project is to find a solution for making all related documents, which may be stored and accessible through various systems, findable in a semantic way. This includes dealing with metadata as well as adding annotations to documents and many other requirements. The approach is called “eThek” (mixture of “e-learning” and “Mediathek” which means “digital library”) and the main goal of this paper is to evaluate which of the existing software solutions on the market meets the needs.

In the past few years, a lot of studies and research were done regarding e-learning software solutions or technology approaches for supporting e-learning requirements. Considering the need for an abstract knowledge model which can be lapped over existing resources, the concept of Topic Maps is mentioned a lot. Topic Maps is a standard for representing knowledge, with an emphasis on the ability to find information. They provide an external meta-structure in the form of a dynamic, semantically based hypertext. Most of the previous studies dealt with the combination of topic maps with classical hypertext-based or concept-based e-learning content such as wiki pages [28]. Providing access to document-based content as learning resources through a topic map driven solution was hardly considered. In this case, the TM4L project [27], which offers Topic Map-based digital libraries, seems to offer a promising approach but is considered by the authors as being in a very early phase of developing.

Besides the idea of a simple Topic Map solution other software solutions like file servers or a combination of them also may be of interest to fulfill the requirements of the eThek platform. Therefore, various software systems were evaluated in this research. One of the main goals of this paper was to find software which provides the most approximate solution and promises as less additional development needs as possible. Therefore, simple engines, stand-alone components or software which is not ready to use, are out of scope. Finally, four software solutions were chosen to be compared within this paper: Microsoft SharePoint, W3C Annotea, Ontopia and Caringo CASTor.

Chapter 2 provides a closer look at the eThek requirements which will be part of the further assessment in chapter 3. After the description regarding components, architecture, and basic functionalities of each of the chosen software, an evaluation follows how the software meets the needs. Every subchapter closes with a possible usage scenario regarding the requirements. An evaluation of all the software solutions considered and an outlook for further projects can be found in the conclusion. For better understanding it is recommended that the reader has basic knowledge about topic maps and related technologies such as XML.

12 REQUIREMENTS

The eThek is defined as a central web application for universities for storing documents that can be accessed by multiple users over a network. Documents stored in eThek include written documents, like course materials, scientific papers and presentations but also multimedia files like videos or music.

Annotation data. It should be possible to add annotations to documents stored in eThek. These annotations should be stored in a central database to allow access for multiple users. Information that must be stored with annotation includes date, username, a title, a rating and the path to the document. The creation of additional data fields for annotations should also be possible. Furthermore, it is important that access to annotations can be restricted.

Import / Export of objects and their annotations in XML. An import/export function for documents and annotations stored in eThek is required, preferably through XML-documents.

Search. The search functionality of eThek must allow searching for documents, within documents and for annotations.

User and Access management. The eThek should be able to integrate existing directory information services over LDAP.

Central document storage. Central document storage is required for all kinds of learning materials and documents in various formats (Microsoft Office, PDF, video, audio, etc.).

Central annotation storage. Annotations should be stored in a way that makes them easily accessible for administrators or authors.

License and restrictions. The preferable license model is open source or a free license for educational institutions.

Support and maintenance. It is important that there is support available for the underlying software of the eThek and that continuous maintenance and further development is ensured.

13 SUGGESTED TOOLS AND TECHNOLOGIES

As mentioned in the introduction, only software solutions were considered, which are ready to use and promise as few additional developments need as possible. Regarding Topic Map concepts, mainly the list of tools from topicmaps.com (a site dedicated to promoting the use of topic maps) [29] was used for the research. The Ontopia Knowledge suite turns out to be the most mature software solution. In addition, three more software solutions are evaluated: Microsoft SharePoint, which provides integrated services to support almost all required features, W3C Annotea, a project which comes directly from the World Wide Web Consortium and is part of the Semantic Web efforts, and Caringo CASTor, a file server solution providing functionalities to store a huge amount of unstructured data.

13.1 Microsoft SharePoint

SharePoint is a Web application platform developed by Microsoft. The current stable release is of the year 2013 and is known for intranet content management and document management. Although, the current version has much more to offer than only content and document management and has some significantly broader capabilities. The new version of SharePoint offers intranet portals, document & file management, collaboration, social networks, extranets, websites, enterprise search, and business intelligence. It also has system integration, process integration, and workflow automation capabilities. Another thing to be mentioned is that the information given in this paper partly comes directly from the Microsoft SharePoint Solution Specialist Mr. Richard Bosse. Therefore, some information cannot be referred to in the literature link, because it was part of the interview process. The interview text is available on demand. Like the other Microsoft products, SharePoint has a Microsoft Office-like interface, and it is closely integrated with the Office suite and is easy to understand also by non-technical users. [1] Furthermore, SharePoint also incorporates a

complete development stack based on web technologies and standards-based APIs. As an application platform, SharePoint provides central management, governance, and security controls for implementation of these requirements. [2] The SharePoint platform integrates directly into IIS (Internet Information Services) - enabling bulk management, scaling, and provisioning of servers, as is required by the UAS Technikum Wien. Figure 1 shows the logical architecture of SharePoint

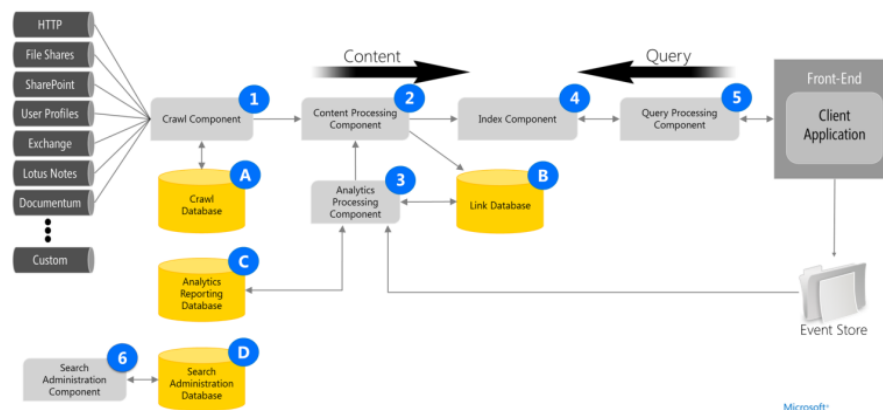


Figure 3: Share Point Architecture [2]

This high-level overview of the new search logical architecture shows all the components of the new search engine. All the content sources are placed on the left side of the picture that are going to be picked up by the new crawl component (1). The crawl component is going to get track of what it needs to crawl and whether it was successful in the crawl database (A). It hands off the content it crawls to the new content processing component - CPC (2). In the CPC SharePoint breaks down what was crawled into the individual pieces of data that will get stored in the index. It also writes to the link database (B) to track all the different anchor content it has crawled and then pushes the data to the index component (4). The index component is what handles the process of creating the physical index, as well as responding to queries. The queries are delivered by the query processing component (5). It takes the individual queries from a client application and figures out the one-to-many queries that are required to execute upon that and then aggregates the results and sends it back to the client. Periodically a timer job will run and perform analytics on the SharePoint content. This includes what content has been accessed through different query results, as well as what content has been accessed by users just clicking in and around different SharePoint sites. It pulls in data from the links database for additional relevance data and stores the reports in the analytics reporting database (C); it also pushes a small amount of analytical data back to the content process component so that it can be included in the index. The search component (6) is what's responsible for keeping it all up and running and healthy, and it stores its configuration in the search admin database (D). [3]

Annotation data. Annotations can be defined individually by any user or admin, uploading a file. It is also possible for UAS Technikum Wien to create their own taxonomy.

Import / Export of objects and their annotations in XML. It is possible to add metadata and store it together with the documents. The documents are stored centrally, and the annotations can be document based. Moreover, it also supports different data formats, such as videos, pictures, links, etc.

The search function is also applicable for the file content and file names. Also, searching for Meta information is possible. However, it is not a standard feature to import/export objects

with annotations in XML, but possible with the help of some migration tools e.g., Microsoft Excel.

Search. SharePoint makes it possible for the admin to define data fields for the manual annotations, such as username, date, datalink, title, and rating and fields for free text. Moreover, it is also possible for the user to use automated annotation, using keywords to search for the documents.

User and Access management. SharePoint offers the possibility to define different roles for the users, like student, teacher, admin, etc. Going one step ahead, SharePoint offers the functionality to administrate user groups (annotation groups) with the help of Active Directory and supporting LDAP. The following Figure 2 shows a preview of the admin area managing users' properties, profiles, sub-types, audiences, permissions, policies and much more:



Figure 4 SharePoint User Management

Central document storage. Another plus point is the central data storage and user capacity, which is huge and offers enough scalability and space for the users and their stored documents. It can be expanded on demand.

Besides using SharePoint only for data storage, it can also offer the additional functionality of personal or Team Websites. Figure 3 gives an example about the SharePoint site collection. A Farm is a collection of physical SharePoint servers that serve the content to users. Generally, such a Farm can be structured and build using different types of topologies.

- Farm: collection of SharePoint Web Applications
- Web Application: collection of SharePoint sites
- Site Collection: Formed by a single root site (Top-Level site) and a collection of sub-sites.
- Sub-Site: collection of pages, lists, libraries, and other content the user wants to make available through the browser. [4]

SharePoint FARM

- **Web Application 1**
 - **Site Collection 1.1**
 - Top Level Site
 - Sub-sites
 - Site 1
 - Lists
 - Libraries
 - Other content
 - Site N
 - **Site Collection 1.2**
 - **Site Collection N**
- **Web Application N**
 - ...

Figure 5 SharePoint FARM

Central annotation storage. Considering annotation storage, SharePoint offers a central place to store the annotations. The following Figure 4 gives an example of the possible implementation and taxonomy hierarchy:

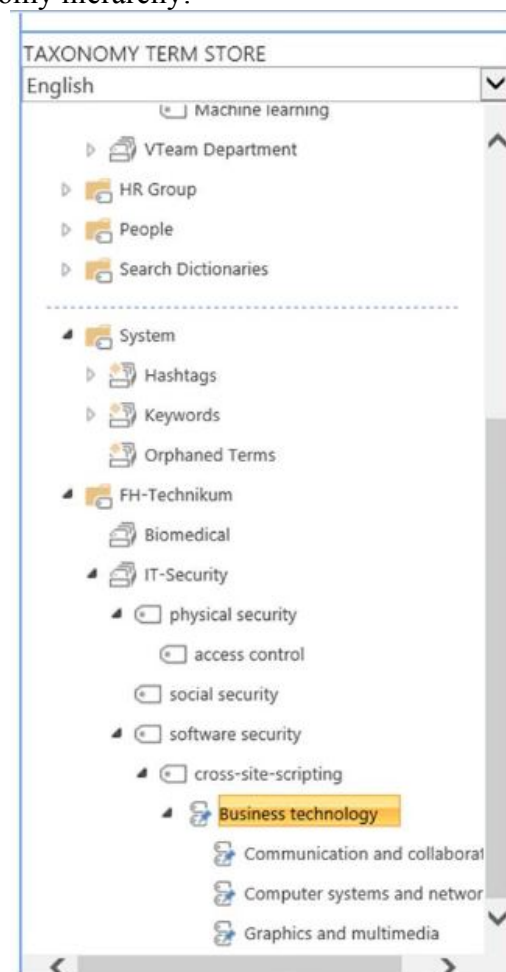


Figure 6 SharePoint Taxonomy

License and restrictions. SharePoint is not an open-source product. But for UAS Technikum Wien, Microsoft has a special academic offer for students and is almost free which just needs to be applied, providing a cloud service edition as part of their Office 365 platform.

Support and maintenance. Considering the support service of Microsoft, one can rely on them. Having a lot of Service Partners, they offer a 24-hour support, which is granted online or via telephone. In urgent cases, it is also possible to call a SharePoint specialist to fix the problem.

Possible usage scenario and evaluation. The Microsoft SharePoint platform provides all the basic needs and ready to use infrastructure needed for several file formats. Furthermore, additional Add-Ons such as better portraying of the Annotations as a Topic Map Format can be bought additionally. The Taxonomy is given, but the way of portraying it might not feed the needs of UAS Technikum Wien, as the Annotations cannot be displayed graphically in a topic map format.

Having a look at scalability, data storage, Business Intelligence, and User capacity, one does not have to worry. Additionally, the Support Function in case of Errors is also provided. On top of that, using this Cloud Service the User does not have to worry about the security and stored data in the cloud. The data stored is in Europe and applies to the Safe Harbor contract. Overall, the software covers most of the necessary functions needed for eThek. With a few additional applications for the annotations and topic map navigation.

13.2 W3C Annotea

Annotea is a W3C project that enhances collaboration via shared metadata-based Web annotations, bookmarks, and their combination. Figure 5 presents the architecture of Annotea. We have various stores for RDF metadata. Metadata can be stored locally or on one or more annotation servers. And we have a user interface that provides different views on the web resources.

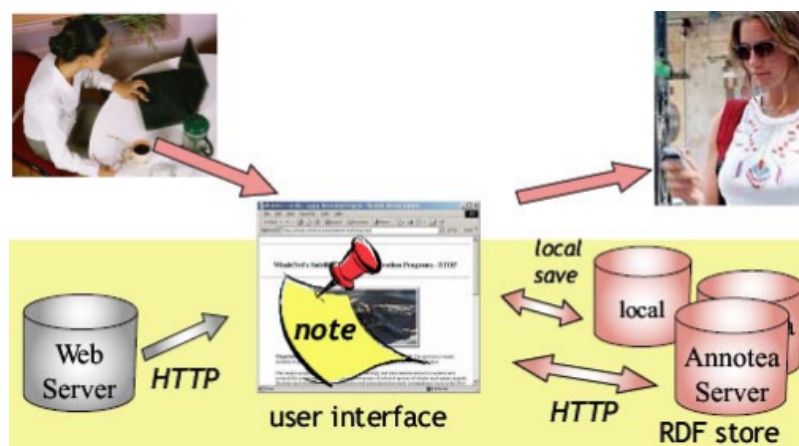


Figure 7 Annotea architecture [33]

Annotation data. Annotations can be comments, notes, explanations or external remarks and are stored as Annotea objects. They can be attached to web documents or a selected part of the document. Annotations are described using the RDF [30] schema and are located in the document by using XPointer [31].

Annotea objects have a URI, containing RDF metadata and a property referring to some other web resource. For instance, annotations can annotate another web page using the property

“annotates”. Additionally, Annotea objects include a core set of properties for describing web resource. For instance, annotations have a “creator” property for describing the creator of the object [34].

Import / Export of objects and their annotations in XML. Annotea can only be used for web documents which can be identified by URI. The annotations are assigned to that URI and are stored in annotation servers using the RDF schema. There is no export functionality but the server uses MYSQL and ODBC so it should be possible to develop an export functionality [34].

Search. The presentation of the annotations, including the search functionality, can be done using a web browser. Every time the user browses a document, the browser queries the defined annotation server, requesting the annotation metadata that is associated with the document’s URI. Annotea does not include a search functionality for annotations, therefore the user cannot search for specific annotations. In fact, annotations will only be displayed after opening the document’s URI [34].

User and Access management. The Annotea project does not include user management.

Central document storage. Documents and resources are associated with a unique URI and can be stored on a webserver.

Central annotation storage. The annotations attached to a document are stored locally or in one or more annotation servers and can be loaded after the user gets the document. The user can select which form of these metadata stores he or she wants to retrieve the Annotea objects. Users can also select the store for writing new Annotea objects. As shown in the HTTP protocol is used for the interaction between the clients and the annotation server.

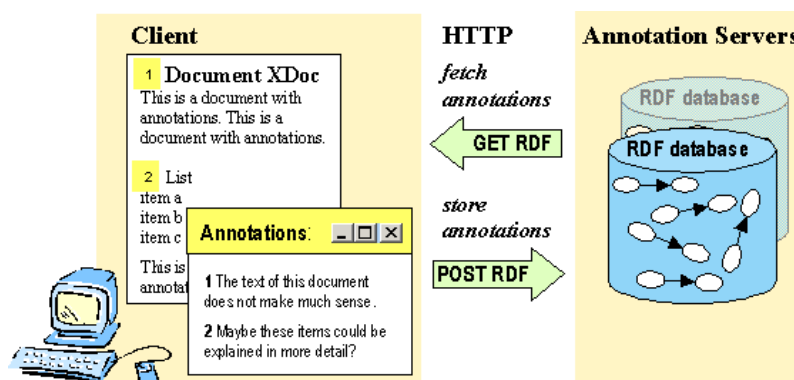


Figure 8 Annotation storage and usage [34]

License and restrictions. Annotea uses the W3C standards when possible and is a full open-source project [35].

Support and maintenance. The last official update for Annotea was published in October 2005 [35].

Due the fact that Annotea is not actively maintained any more it can only provide some conceptual ideas for the eThek project. Furthermore, the source code could be used to develop its own software solution for eThek. There are several related projects with implementations for annotation servers and clients which could be evaluated in additional projects [32].

13.3 Ontopia

The Ontopia product is an open-source project, created by the Ontopia community. It was first released in 2001 and in November 2012 version 5.2.2 was available. A little number of commercial projects are known to use Ontopia. The product provides a set of tools for building applications driven by topic maps, such as Web portals and knowledge-based intranets and it is used for content management systems as well as for enterprise application integration. [11]

Ontopia consists of the following components known as Ontopia Knowledge Suite:

- Ontopia Topic Maps engine (full-text integration, schema tools, query engine)
- DB2TM (database conversion)
- Ontopia Navigator framework
- Ontopia Web editor framework
- Ontopia RDBMS Backend Connector
- Ontopoly Topic Maps editor
- Ontopia Omnigator (Topic Maps browser)
- Ontopia Vizigator (visualize and navigate topic maps)
- Figure 7 shows how the components work together.

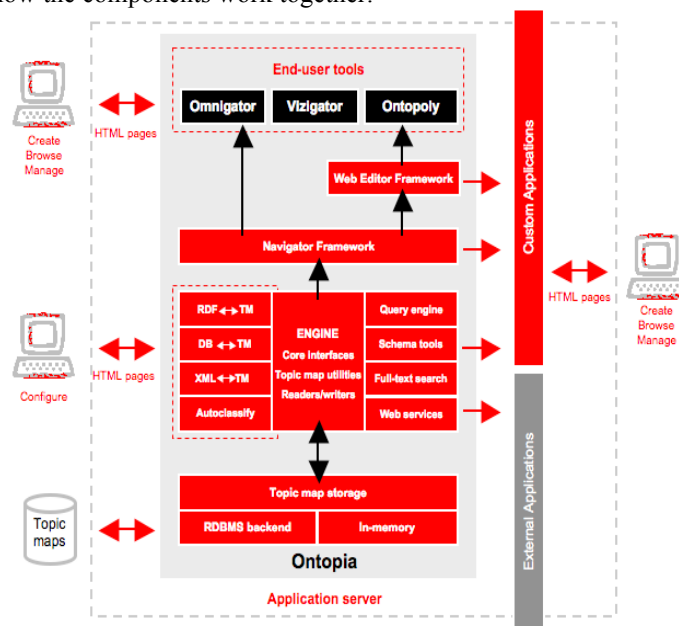


Figure 9 Ontopia Knowledge Suite [12].

The Topic Maps engine is the core component of Ontopia's Knowledge Suite which stores and maintains the topic maps. Essentially it contains a set of Java APIs providing services for accessing and modifying topic maps, as well as importing, exporting and converting services. Tolog (query language for topic maps) is used for query execution and full-text search is performed. The engine can store topic maps in relational databases or keep them in memory. In both cases the API to the engine stays the same. [13]

Using DB2TM ("database to topic maps") module relational data can be automatically converted to Topic Maps data. A simple XML file is used to describe the mapping. The module also provides synchronizing to keep the topic map up to date as changes occur in the database. [14]

The RDBMS Backend Connector adds relational database persistence support to the Ontopia Topic Maps engine while the persistence is transparent for the user of Topic Map interfaces. [15]

With the Navigator Framework it is possible to create individual topic map-based websites. It mainly provides a set of tag libraries and produces no HTML itself in order to have complete control over the layout and structure of the desired web application. It is based on J2EE (Java 2 Enterprise Edition) platform and uses JSP (Java Server Pages) technology which is easy to learn and efficient in use. [16] The Web Editor Framework is an extension of the Navigator Framework. It is based on JSP technology and provides extensive libraries of actions for developing web applications where the users can modify the contents of a topic map using a forms-based interface. [18]

Ontopia offers three end user-tools for editing topic maps based on Ontologies (Ontopoly), for browsing and debugging (Omnigator) and to visualize and navigate through topic maps (Vizigator).

Ontopoly is written as a client/server application and is accessible within Ontopia – as a client you use the web browser while the server application (which is built with the Navigator and Web Editor Framework) is a web server bundled with the distribution. It provides a web interface to design Topic Maps ontologies as well as to populate them with instance data. It is divided into the ontology editor and the instance editor. An administration interface makes it possible to add metadata to the topic map (description, version number, etc.), validate it and provide functions for importing and exporting from and to various interchange systems. [18] The Omnigator (“omnivorous navigator”) application (client/server) is the main Topic Map navigator for loading and browsing topic maps and is built using components of the Ontopia Knowledge Suite (f.e. Navigator Framework). Although it is provided as an end-user tool, the purpose of the application is more like debugging for (topic map) developers. It is intended as a teaching aid, to help understanding Topic Map concepts. [19]

While the Omnigator uses a text-based interface the Ontopia Vizigator provides an additional method of topic map navigation through a graphical interface. It consists of three components which are dependent on each other: the designer’s configuration tool (VizDesktop), Java applet for adding visualization to web applications (VizLet) and a Plugin for combining text-based view and graphical-based view within the Omnigator (VizPlugin). Vizigator naturally bases on TouchGraph which is an open-source toolkit for graph visualization (Sourceforge). VizDesktop (desktop application), can be seen as the entry point to the entire Vizigator. With VizDesktop designers can determine whether and/or how various topic types and associations will be graphically displayed. All the configurations that are done with VizDesktop (filter settings, shapes, sizes, colours, fonts, etc.) are stored at the server and then loaded from the other components when they are used. Thus, VizLet provides almost the same visualization functionality as VizDesktop, except that the menu bar is unavailable, and it always operates in topic view (while VizDesktop also shows ontologies). It is a Java Applet (reusable applet component) which is intended to add topic map visualization to any web application using Ontopia. VizPlugin is an Omnigator Plugin that provides graphical visual navigation of topic maps and uses VizLet to support this visualization directly in the browser. [20]

Annotation data. Regarding the given requirements, topic maps can be used for structuring and saving document properties and metadata in the desired way. While the data structure is provided by ontology, the topics within the topic maps depict the documents containing the predefined fields from the ontology. Besides topic types, names, associations, roles and so on, occurrences can be added which mark the document properties. Various data types are provided (String, URI, Date, Datetime, Number, as well as HTML content and images).

In addition, Ontopia provides an automatic classification module which helps creating a topic map (or simply classifying the content) by offering extracted keywords from a given file, which could be an XML, HTML, PDF, Microsoft Word or Microsoft Powerpoint document in English and Norwegian language. Regarding Ontopia's product description [21] additional file formats and languages can be plugged in easily.

Unfortunately, adding metadata (annotations) to documents is hardly possible with the given functionalities – no solution was found to add files to the topic maps and neither for browsing them in the graphical view. It can be assumed that Ontopia was designed much more to combine with content management systems for web content solutions than to use topic maps together with document management systems.

Import / Export of objects and their annotations in XML. Non-Ontopoly topic maps (maps from an external source) can be imported and will automatically convert according to the most approximating topic map in the system. Additional user support may be needed.

Ontopia uses the ontology topic maps as system topic maps which are needed for understanding and interpreting topic maps within the system in the right way. Exporting topic map instances usually work without the underlying ontology, but system topic maps must remain. [18]

Omnigator allows loading an RDF model and browse it as it were a topic map, and exports it in XTM syntax. Also export a topic map as an RDF model is supported as well as merging features. [19] XML support is provided for converting in every direction.

Search. Searching within the system is mainly provided by the Ontopia Omnigator which is responsible for navigation. The Full-text Search Plugin allows doing simple full-text searches when a full-text index has been created first. Combined with the Omnigator's browsing interface it makes the task of locating information much simpler, because jumping directly to a specific topic is possible. Unfortunately, this works only for topic maps data, not for searching within (external) documents. [19] Furthermore, the graphical visualization of topic maps is not suitable for navigation because it doesn't work for all kinds of data.

User and Access management. If building an application which doesn't use directory information stored elsewhere (or uses no authentication at all) is considered, Ontopia provides an application that can be used to administrate topic maps containing users, roles and privileges. Written using the Web Editor Framework it provides user vocabulary understandable by JAAS (Java Authentication and Authorization Service) which authenticate the users against the information in the topic map. [22]

Central document storage. Document storage and management functionalities (regarding external documents beside topic maps, e-learning documents) are not provided by Ontopia.

Central annotation storage. Considering topic maps as pool for all given annotations, Ontopia provides mechanisms for saving them in a relational database (DB2TM, RDBMS) or as files on a server which then marks the central annotation storage. However, a file server or database technologies are not provided by Ontopia Knowledge Suite. But the product supports various databases like Oracle 8 or newer, Microsoft SQL Server 2005 or newer, PostgreSQL 7.4 or newer, MySQL 5.0 or newer, and h2 version 1.1 or newer. [13]

License and restrictions. Ontopia is 100 % Java and runs on any operating system which has Java 1.5. It is a fully open-source product hosted by Google Code. [23] Beside the Apache License 2.0 policies no restrictions regarding number of users and annotations (topics and topic maps) are known. [10]

Support and maintenance. So far known, commercial support is provided by a few companies in Norway, Japan and Holland, which is offered by them mainly as consultant services. [24] Thus, support for individual projects can hardly be considered as it is typical for using open-

source software. As a result, the maintenance workload depends on the designed application's architecture and the way, topic maps are used in it.

Possible usage scenario and evaluation. The Ontopia community describes the Ontopia Knowledge Suite as the world's most feature-complete Topic Maps product. [25] This might be true, but it is far away from offering a fully integrated service application. Ontopia can be seen as a "box of bricks". A lot of development work is needed in order to reach the individual project goals. A solution for adding the metadata (annotations stored as topic maps) to the documents through any kind of document management tooling must be found, database technology must be evaluated, designed, implemented and continuously maintained and a front-end application must be created which may adopt an existing user management solution. In turn, the product's components are very well documented (developer's guides, user manuals) and can be marked as high integrate-able (end-user tools like Ontopoly, Java technology). Therefore, reusing some of the tools and functionalities (especially for creating, storing and maintaining topic maps) may be considered when putting an e-learning concept into practice.

13.4 Caringo CASTor

Caringo CASTor is a proprietary Object Storage Software developed by Caringo, Inc; based in the USA. It is used for storing huge amounts of unstructured data. The minimum hardware requirements are an x86 server with at least 1 GB RAM, a hard drive and Gigabit Ethernet. Multiple Servers can be combined with a cluster [5].

Each document that is stored in Caringo gets a Universally Unique Identifier (UUID) that can be used as a reference to the document in a database, an application or another document. Application integration is possible over an HTTP-interface with the Simple Content Storage Protocol (SCSP) [6].

Several modules can be used to provide additional functions and services like indexing or additional data protection. The Caringo CASTor Content File Server (CFS) allows the creation of interfaces to other applications for file transfers with protocols like CIFS, NFS, FTP and WebDAV [7].

Potential customers can apply for a test account through the homepage [8]. For evaluating Caringo CASTor for eThek the authors applied for a test account. Unfortunately, this test account was not activated by Caringo. Since no other reliable information on CASTor could be obtained by the authors, the information presented in this chapter is entirely based on the information provided by Caringo on their homepage.

Annotation data. CASTor provides default metadata for stored documents. According to the FAQs [6] it is possible to add additional Metadata fields. Therefore, it should be possible to add annotations as metadata.

Import / Export of objects and their annotations in XML. Information on Import/Export functionality for metadata could not be obtained.

Search. The Indexer module for CASTor provides "real time indexing" and "ad-hoc searches of content within CASTor by name or metadata [10].

User and Access management

The CASTor Content File Server module supports integration with Active Directory and Access Control Lists [9]. The requirements for eThek are therefore fulfilled.

Central document and annotation storage. CASTor can be used to store documents and metadata like annotations in a central storage platform and would therefore fulfill this requirement [6].

License and restrictions. CASTor is proprietary software. License information is not freely available for potential customers.

Support and maintenance. Since CASTor is proprietary software, support is provided by Caringo. Caringo provides support 24/7. A knowledge base is also available. Customers can gain access to additional functions like creating trouble tickets and downloads after logging in [11].

Possible usage scenario and evaluation. The Caringo Object Storage Platform provides basic infrastructure for storing documents. Additional modules that should be considered for the eThek are the Content File Server for additional interfaces to other applications and the Indexer for search functionality.

The Caringo Object Storage Platform could be used to store all documents for the eLearning 3.0 platform. Additional software would be necessary to provide functionality for annotations and topic map navigation.

Therefore, the software only covers part of the necessary functions for eThek, and additional applications and programming would be necessary for annotations, topic map navigation and document access. Figure 8 shows a possible system landscape.

Caringo CASTor is used in combination with the Indexer-Module for storing documents in the eThek. The Content File Server (CFS) is used as an interface for the Learning Management System (LMS) Moodle that is currently used at the UAS Technikum Wien. Moodle uses CASTor as document storage and provides the user with access to the documents. To be able to integrate Moodle and CASTor would require additional programming.

For annotations another application would have to be used to provide the necessary functionality for eThek, like topic map navigation. A possible choice could be Ontopia. However, additional programming would be necessary to integrate it with Moodle and CASTor. Another possibility would be to develop new annotation-software according to the requirements.

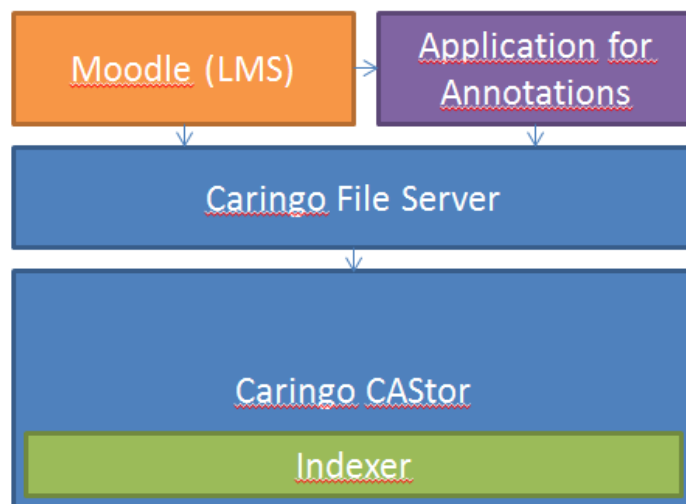


Figure 10 Usage Scenario for Caringo Castor in eThek **Error! Reference source not found.**

Since no test account for Caringo CASTor and the Content File Server could be obtained for this evaluation, all information regarding Caringo CASTor in this paper is from the official Caringo homepage and marketing materials, since no reliable independent sources could be found.

The software uses a proprietary licence. Information on licensing and pricing are only provided after request.

Additional evaluation, additional information about pricing and the license model and a test account are necessary if the Caringo CAStor Object Storage Platform is considered as a viable option for the eThek. However, as it uses a proprietary license and does not fulfill important requirements for the eThek like topic map navigation and annotations it can only be used in combination with other applications. The Content File Server should facilitate integration with other software.

14 CONCLUSION

Table 1 gives an overview of the evaluation described in the previous chapters. The values can be understood as follows:

- (+) Positive; The Software provides all essential services and functionalities. Only implementation and configuration work are needed.
- (~) Caution; the solution provides only approaches where additional programming is required.
- (–) Negative; the solution doesn't provide services or functionalities to support the required features at all.
- (?) Unknown; the necessary information for evaluating a solution regarding the requirement could not be obtained.

Table 18. Tool evaluation

	MS SharePoint	W3C Annotea	Ontopia	Caringo CAStor
Annotation data	~	+	+	~
Import / Export	~	~	+	?
Search	+	-	~	+
User and Access Management	+	-	~	+
Document storage	+	~	-	+
Annotation storage	~	~	+	~
License / Restrictions	+	+	+	-
Support / Maintenance	+	-	-	+

After the evaluation of the selected products, it may be asserted that none of the products is able to meet all requirements on the eThek project. The probed products hold various strengths and weaknesses, which may be looked at from different points of view. While Microsoft is providing a, even though for the concept of eThek not complete, but for itself

functioning and solid version, the other three products require substantial effort to be programmed, and accordingly a combination of different approaches to meet the respective demands.

Whilst Annotea, respectively Ontopia are less suited for productive use on grounds of scarce support, Sharepoint and Caringo are continuously being enhanced and improved. Annotea and Ontopia are both being operated under the OpenSource license and are hence able to provide valuable assistance in terms of independent development of the eThek project as opposed to the other two products. Further strengths of the commercial products Sharepoint and Ontopia lie in the range of searching and of an existent User Mangement.

Based on the conducted evaluation and the results, as displayed in Table 1 Microsooft Sharepoint complies with most placed requirements and furthermore provides the possibility of a flexible use and an extension to the software. Additional strengths of Sharepoint as opposed to other tested solutions would be constituted by the centralized document storage, the commerce with a variety of file types, and the employment of a software solution that will also be actively enhanced and maintained in future times.

In case the University of Applied Sciences Technikum Wien is considering an in-house development, both softwares Annotea and Ontopia would be of relevance. Whereas in case a preferably complete and solid solution shall be employed, Microsoft Sharepoint would depict the program of choice. Because Sharepoint was not conceptualized as sheer annotation software though, it does not offer out of the box a possibility for development of Topic Maps. In that case an evaluation of how Sharepoint could be amplified with annotation functionality as well as Topic Maps would be necessary.

REFERENCES

- [45] Microsoft, SharePoint Overview Microsoft Corporation.
- [46] Microsoft, SharePoint Evaluation Guide Microsoft Corporation.
- [47] MSDN Blog, SharePoint is here,
http://blogs.msdn.com/b/sharepoint_strategy/archive/2012/07/16/sharepoint-2013-is-here.aspx
- [48] Micheltriana, SharePoint architecture, <http://micheltriana.com/2010/03/01/sharepoint-architecture/>
- [49] Caringo, Inc. CAStor Data Sheet,
<http://www.caringo.com/downloads/datasheets/Caringo-CAStor-Object-Storage.pdf>
- [50] Caringo, Inc. Frequently Asked Questions,
<http://www.caringo.com/resources/faqs.html>
- [51] Caringo, Inc. Caringo CAStor, <http://www.caringo.com/products/CAStor.html>
- [52] Caringo, Inc. Register for Caringo Connect,
<https://connect.caringo.com/2.0/user/register>

- [53] Caringo, Inc. Content File Server, <http://www.caringo.com/products/content-file-server.html>
- [54] Caringo, Inc. Caringo Indexer, <http://www.caringo.com/products/indexer.html>
- [55] Caringo, Inc. Caringo Support System, <http://www.caringo.com/support.html>
- [56] Ontopia, About Ontopia, <http://www.ontopia.net/page.jsp?id=about>
- [57] Ontopia, Installation and getting started, Version 5.1,
<http://www.ontopia.net/doc/current/install.html>
- [58] Ontopia, The engine, <http://www.ontopia.net/page.jsp?id=engine>
- [59] Ontopia, Database conversion, <http://www.ontopia.net/page.jsp?id=db2tm>
- [60] Ontopia, The RDBMS Backend Connector Installation Guide, Version 5.1,
<http://www.ontopia.net/doc/current/rdbms/install.html>
- [61] Ontopia, The Ontopia Navigator Framework, Developer's Guide, Version 5.1,
<http://www.ontopia.net/doc/current/navigator/navguide.html>
- [62] Ontopia, The Ontopia Web Editor Framework, Developer's Guide, Version 5.1,
<http://www.ontopia.net/doc/current/webed/devguide.html>
- [63] Ontopia, Ontopoly: The Topic Map Editor, User's Guide, Version 5.1,
<http://www.ontopia.net/doc/current/ontopoly/user-guide.html>
- [64] Ontopia, Omnigator: The Topic Map Browser, User's Guide, Version 3.4,
<http://www.ontopia.net/omnigator/docs/navigator/userguide.html>
- [65] Ontopia, The Ontopia Vizigator, User's Guide, Version 5.1,
<http://www.ontopia.net/doc/current/vizigator/userguide.html>
- [66] Ontopia, Automatic classification, <http://www.ontopia.net/page.jsp?id=classify>
- [67] Ontopia, User Management in the OKS, Developer's Guide, Version 5.1,
<http://www.ontopia.net/doc/current/navigator/userman.html>
- [68] Ontopia, Developer Site for the Ontopia project, <https://code.google.com/p/ontopia/>
- [69] Ontopia, Services, <http://www.ontopia.net/services.jsp>
- [70] Ontopia, Learn, <http://www.ontopia.net/learn.jsp>
- [71] TM4L Project, http://iiscs.wssu.edu/drupal/tm4l_welcome
- [72] Using Topic Maps for E-Learning, <http://www.win.tue.nl/~laroyo/Conf-2003/CATE2003/CATE2003-dichev-dicheva-aroyo.pdf>
- [73] List of Topic Map tools by topicmap.com, <http://topicmap.com/tools>

- [74] RDF Working Group, Resource Description Framework (RDF),
<http://www.w3.org/RDF/>
- [75] Henry S.Thompson, W3C XML Pointer, XML Base and XML Linking,
<http://www.w3.org/XML/Linking>
- [76] Marja-Riitta Koivunen, Annotea related projects,
<http://www.w3.org/2001/Annotea/Projects#servers>
- [77] Marja-Riitta Koivunen, Ralph Swick, Eric Prud'hommeaux, Annotea Shared
Bookmarks, <http://www.w3.org/2001/Annotea/Papers/KCAP03/bminfra.png>
- [78] José Kahan, Annotea: An Open RDF Infrastructure for Shared Web Annotations,
<http://www10.org/cdrom/papers/488/index.html>
- [79] Marja-Riitta Koivunen, Annotea Project, <http://www.w3.org/2001/Annotea/>