Tools and Services for Knowledge Discovery, Management and Structuring in Digital Libraries

Uffe Kock Wiil
Department of Computer Science
Aalborg University Esbjerg
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
ukwiil@cs.aue.auc.dk

David L. Hicks
Department of Computer Science
Aalborg University Esbjerg
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
hicks@cs.aue.auc.dk

Abstract

This paper presents a component-based open hypermedia environment for supporting scholarly work processes such as associative storage and retrieval, information analysis, and classification in digital libraries. In particular, a scenario illustrating the visions of the Construct environment is presented together with a description of the basic tools and services for knowledge discovery, management, and structuring in digital libraries. Presently, the tools and services support most of the scholarly work practices presented in the scenario.

Keywords: digital library, scholarly work, hypermedia structuring mechanisms, open hypermedia, Construct.

1 Introduction

The ongoing shift from traditional libraries to digital libraries is well on its way and has resulted in many new challenges as well as many new possibilities.

The challenges are to support standard library work practices such as storing, cataloging, searching, accessing, sharing, and disseminating information in a digital setting where different types of documents are available over a network from large, distributed data sources [9, 23].

Having documents available online in electronic form opens up several new possibilities with respect to support for knowledge management work practices. In this paper, we will describe a set of tools and services that support work practices related to scholarly work [31]. In particular, we present an environment named Construct that provides different mechanisms for knowledge discovery, management, and structuring.

Many researchers have argued the need for hypermedia structuring mechanisms in digital libraries (e.g., [8, 18, 20, 27]). However, many of these researchers focus on the provision of associative (linking) structures only. The Construct environment provides support for additional types of structuring mechanisms that are useful in scholarly work settings such as spatial and taxonomic.

The remainder of the paper is structured as follows. Section 2 presents a scholarly work scenario illustrating the visions and goals of the Construct environment. In Section 3, we present the Construct environment with a focus on the tools and services that support knowledge discovery, management, and structuring. Section 4 outlines future work, and Section 5 concludes the paper.

2 Scenario

This scenario illustrates the visions and goals behind the previous, present, and future work with the Construct environment with respect to providing support for knowledge discovery, management, and structuring in digital libraries.

2.1 Application domain

The scenario is a typical scholarly work scenario where a scholar (e.g., a student or a researcher) is trying to get an overview of a new topic (application domain). A large amount of literature is available on the topic, distributed across many documents, which makes it difficult to get a quick overview. The scholar uses a combination of existing digital library tools and services and Construct tools and services to assist in the knowledge discovery, management, and structuring processes.

The scenario uses “open hypermedia” as an example of a new topic that a scholar wishes to become familiar with (see [1, 11, 12, 22, 34] for examples of prominent open hypermedia systems). The scenario will show how the Construct tools and services can assist the scholar with organizing and understanding the literature in the open hypermedia domain.

2.2 Digital library setting

The scenario takes place within the context of the ACM SIGMOD digital library, which is a service that is provided to ACM SIGMOD members. The digital library contains a large collection of papers from database and database related conferences and journals. The digital library currently consists of two parts: the “Anthology” comprising 19 CD-ROM’s and the “Digital Symposium Collection” comprising 4 CD-ROM’s. The documents from all these CD-ROM’s have been placed online on a central server for easy and fast access. The current set of documents and indices in the ACM SIGMOD digital library takes up 13.4 GB of disk space (almost 250.000 files). The document set will continue to grow as new CD-ROM's arrive every year. The ACM SIGMOD Anthology is also available on the WWW [16].

The digital library mainly consists of two types of documents: indices in HTML and documents in PDF. Using the available search and browsing tools, users can query these documents in different ways. One way is to search for authors using the DBLP (Digital Bibliography and Library Project) indices – the result is a list of publications available in the digital library by that author. The DBLP indices are distributed as part of the ACM SIGMOD digital library. The most recent version is available on the WWW [17]. Another way is to perform a full text search in the PDF documents – the result is a list of matches for the queried term. The ACM SIGMOD digital library contains PDX index files that allows full text searches to be performed using the Adobe Acrobat software. Yet another way is to browse the documents by following the links on the HTML pages using a standard browser.

2.3 Construct tools and services

The scenario makes use of the following Construct services:

- The metadata service, which allows scholars to add arbitrary metadata (consisting of key-value pairs) to all types of documents in the digital library.
- The navigational structure service, which allows scholars to add arbitrary associations (links) between all types of documents in the digital library.
- The spatial structure service, which allows scholars to organize documents on a desktop using a spatial metaphor (similar to organizing documents on a table).
- The taxonomic structure service, which allows scholars to organize (classify) documents in classification trees according to certain criteria.
- The data mining service, which allows scholars to discover automatically different types of relations among documents, based on different input criteria. For example, the data mining service can produce a classification of documents based on their metadata keys and values.

These services have been made available in standard desktop applications such as Netscape 6, Microsoft Word, and GNU Emacs.

2.4 Scholarly work scenario

Anna is a graduate student in computer science. As part of her enrollment in a hypertext course, she is given the assignment to make a report and a presentation to the class about the open hypermedia research area. The students have access to a number of both traditional paper based and electronic literature sources. Anna decides to base her study on the ACM SIGMOD digital library, which is available online in the computer science department and, which, she is told, contains the proceedings of both the ACM Hypertext and ACM Digital Libraries conferences and the ACM TOIS journal (all well known sources for high quality hypertext papers).

ACM SIGMOD digital library browsing. Confronted with her task, Anna decides to start browsing the digital library for open hypermedia papers using a standard browser. After spending some time locating the HT and DL proceedings, she starts looking through these proceedings for papers dealing with different aspects of open hypermedia. She soon realizes that using a sequential browsing strategy will not allow her to solve her task in the allocated time.

ACM SIGMOD digital library querying. Instead, Anna decides to use the full text search capabilities of the digital library and search for the term *open hypermedia*. She also decides to search for the related term *open hypertext*. These queries result in many hits. After quickly looking over these papers, Anna finds several author names that appear on many of the open hypermedia papers. She decides to search for additional papers by these authors in the digital library using the DBLP indices. This results in several new hits that are relevant to her task. Anna now believes that she has gathered enough information to start the process of getting an overview and organizing the papers. Thus, it is now time to deploy the Construct tools and services.

Construct navigational and metadata services. When Anna starts the task of going through all the papers, she discovers that students from past hypertext classes have already added associations (links) between some of the documents and added metadata about some of he documents. By browsing these associations and looking at the metadata, she is able to find more material and understand more connections than would have been possible in the same amount of time using the documents themselves. She adds her own associations between some documents and adds additional metadata about some of the documents.

Construct spatial service. At this point, Anna has some initial and partial understanding of the open hypermedia area, but still lacks a clear overview and fundamental understanding of the problem domain. Anna decides to use the spatial service to help analyze and improve her understanding of the information that she has gathered. She organizes the open hypermedia research papers on the 2D space according to different criteria. Over time, this process reveals relationships between papers and their associated topics that she had not seen before, significantly enhancing her understanding of the problem domain.

Construct data mining service. At this point, Anna knows of many of the central people, many of the prominent systems, and many of the important issues that are involved in open hypermedia research. She decides to structure her findings in the report according to these three different perspectives (central people, prominent systems, and important research issues). Anna now decides to use the data mining service to help generate classifications of research papers based on these three perspectives. The first step is to set up the parameters for the data mining classification algorithms. Anna uses a combination of author names, keywords, Construct metadata keys and values, and Construct link structures to guide the classification algorithms. The next step is to run the classification algorithm, which results in the generation of a classification tree.

Construct taxonomic service. Anna uses the taxonomic service to view the automatically generated classification tree structure. She is not completely satisfied with the result and decides to make some manual changes to the classification tree to match her personal understanding of the problem domain.

The last two steps (automatic generation and manual personalization of a classification tree) are repeated three times to generate the three different perspectives of the open hypermedia research literature. Anna is now ready to write her report and make her presentation to the hypertext class.

3 Construct

The purpose of this section is to describe the ongoing work in the Construct project towards providing support for different types of scholarly work practices in digital libraries – like the scenario outlined in Section 2. This section gives an overview of the Construct environment and provides a description of the previous work and current status of the individual tools and services of the environment. Future plans are described in Section 4.

3.1 System overview

Construct is a component-based open hypermedia system [32] developed at Aalborg University Esbjerg, Denmark [5]. The Construct environment (Figure 1) consists of different categories of software components [30, 32]. The boxes inside the large circle represent the user-oriented tools (applications) and services (explained in Section 3.2).

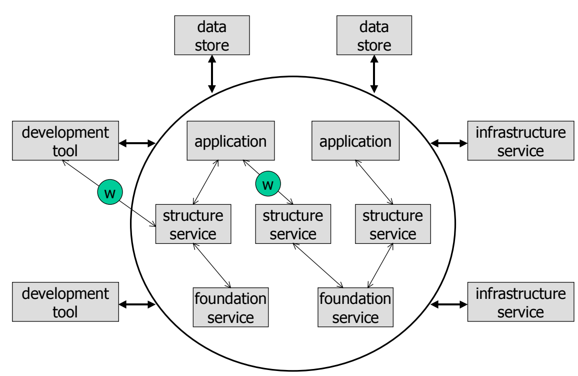

Figure 1. An overview of the Construct environment. The small circles containing a "w" indicate a wrapper service.

- *Applications (tools)*. This category includes desktop applications (e.g., Netscape, Microsoft Word, and GNU Emacs) that have been integrated (modified, extended, or wrapped) to be able to make use of the Construct structure services.
- *Wrapper services.* A wrapper is a service that allows a legacy application to be integrated with

the Construct environment. Natively compliant applications do not need wrappers – all other applications do.

- *Structure services*. Integrated applications can make use of different types of structure services to organize and structure documents located in data stores. Each type of structure service provides a different set of structural abstractions (e.g., navigational, metadata, spatial, taxonomic, argumentation, data mining, workflow, and cooperation) that supports different application domains.
- *Foundation services.* Foundation services provide the very basic services in the environment that most other services depend on. Example foundation services are:
  - *Structure stores*. A structure store is a software component that manages storage and retrieval of structure. Structure stores can handle different types of structural abstractions.
  - *Multiuser and collaboration services.* This category includes services such as concurrency control, notification control, and access control.
  - *Versioning services.* Services that allows structural abstractions to be versioned.
- *Data stores*. A data store could be a file system, a database, a CD-ROM, a web server, etc. Documents in data stores are not altered by the use of Construct services. All structures are stored in the structure stores and are superimposed on the documents at display time by the applications.
- *Infrastructure services*. This category includes general services that enable the individual components of the environment to co-exist. Examples of these services are service discovery, naming, and location.
- *Development tools*. The development tools assist in the development of new services in the environment. Currently, three development tools exist:
  - *UML Tool*. This tool allows the developer to specify new services by drawing UML diagrams in a graphical user interface. UML Tool translates the UML diagrams into IDL specifications.
  - *Emacs.* This well-known editing tool is used to create IDL specifications and Java code as well as documentation of various kinds. Emacs is considered a development tool of the Construct environment due to its integration with the navigational structure service.
  - *Construct Service Compiler (CSC)*. The CSC takes IDL specifications as input and generates service skeletons in Java. Service skeletons are wrapped in components that can automatically run as part of the Construct environment. The developer must add some code to the skeletons to define the semantics of the generated operations.

  Development tools can also be clients of structure services. For example, UML Tool and Emacs are clients of the navigational structure service. Thus, it is possible to link from classes or fields in the UML diagrams to, for example, design rational located in a text file under the control of Emacs.

The Construct environment is based on well-known technologies for building component frameworks including Java Beans and RMI [15]. The entire system is coded in Java.

3.2 Tools and services

Figure 2 gives an overview of the Construct tools and services that either have been developed or are under active development. The remainder of this section describes each of these tools and services in more detail.

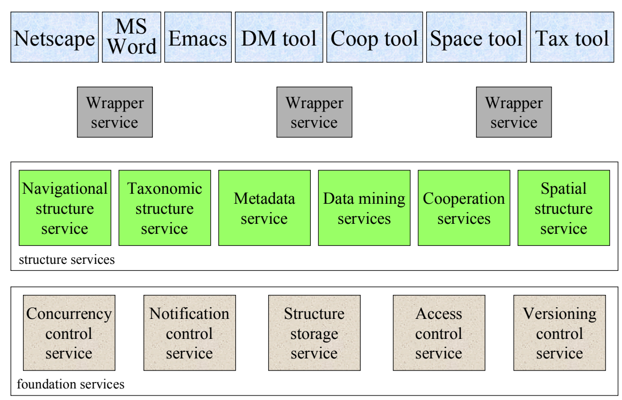

Figure 2. An overview of tools and services in the Construct Environment.

3.2.1 Navigational structure service: The navigational structure service supports associative storage and recall, which is a structuring mechanism (proposed by Vannevar Bush back in 1945 [3]) that allows users to associate arbitrary pieces of information and navigate over these associations. Since then many navigational hypermedia systems have been implemented, starting in the early 1960's (e.g., NLS [7]) and continuing through today (e.g., the WWW [2]). The Construct navigational structure service has been integrated into a number of applications including Microsoft Word, Emacs, UML Tool, Vital [29], and Netscape 6.

The UML Tool provides an example of a typical integration of the navigational structure service into an application. UML Tool was integrated with the Construct navigational structure service using a wrapper written in Java. Figure 3 shows a screen dump of UML Tool when running as an application of the Construct navigational structure service. A new menu (named Construct) with five menu items (Traverse link, Start link, Create anchor, Delete anchor and End link) is added to the interface. This is the only visible indication that UML Tool has been extended with navigational hypermedia services.

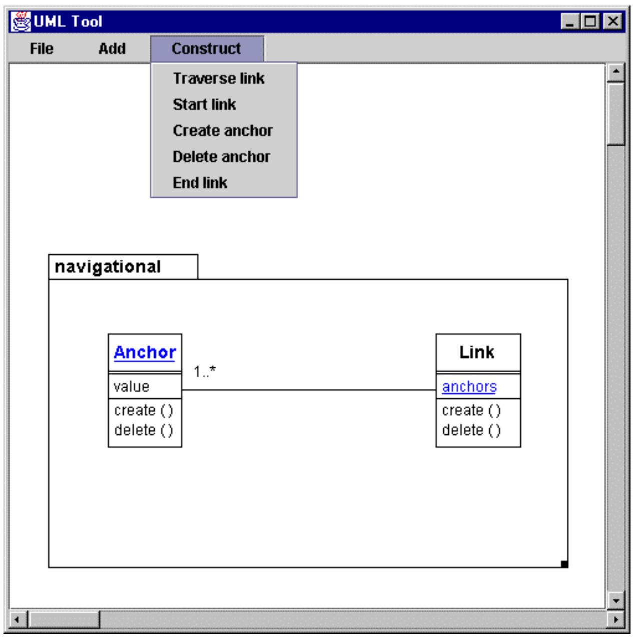

Figure 3. UML tool as an integrated application of the Construct navigational structure service. Anchors are displayed on screen as blue, underlined text.

Users of UML Tool and other integrated applications can navigate existing associations by selecting the anchor marker and activating the *Traverse link* menu item. The result of this action is that the other endpoint of the association will be displayed by its native application and the anchor marker belonging to the traversed link will be highlighted. For example, traversing the link from the Anchor class in the UML diagram in Figure 3 (represented as underlined text) will result in Emacs displaying a file that contains the design rationale for the Anchor class (with a piece of text highlighted to indicate the corresponding anchor point). Users can create new associations by activating the *Start link* menu item. The next step is to select the first endpoint of the link (by selecting a class name or field name in UML Tool) and activate the *Create anchor* menu item. Then the other endpoint must be created. This could for instance be inside an HTML file displayed by Netscape 6. Again, the user selects the endpoint of the link (by selecting a piece of text in Netscape 6) and activates the *Create anchor* action. The final action is to activate the *End link* action from either of the tools. Now the link is stored in the structure store and is ready to be traversed.

As illustrated above, the navigational structure service is able to support all the linking actions performed in the scenario in Section 2.4.

3.2.2 Metadata service: The metadata service supports the addition of a metadata record consisting of an unlimited number of key/value pairs to arbitrary documents. Traditional libraries have used this mechanism (in a paper based form) for many years to help categorize their library materials. The feature is also standard in digital libraries [13, 14]. The metadata service has so far been integrated into Emacs and Netscape 6.

Netscape 6 provides an excellent example of a typical integration of the metadata service into an application. Like UML Tool, Netscape 6 is integrated using a wrapper written in Java. The metadata service is available in Netscape 6 as a sidebar. The *main view* of the metadata service sidebar is shown in Figure 4. The main view supports *Add*, *Update*, *Delete*, and *Find* operations on key/value pairs in the metadata record.

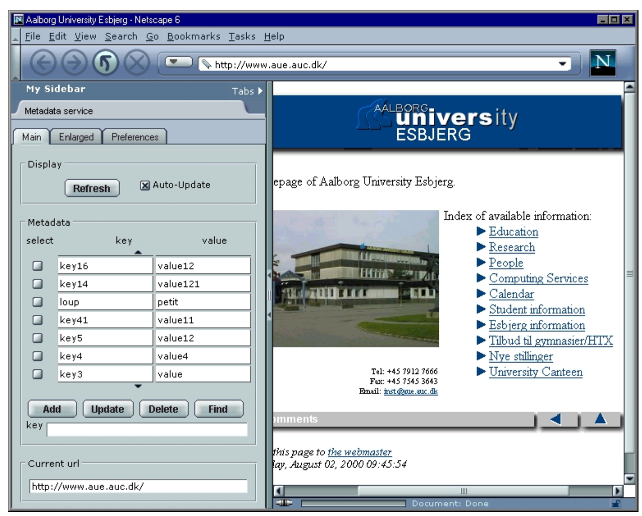

Figure 4. The main view of the Construct metadata service sidebar in Netscape 6.

The *enlarged view* allows a selected metadata key/value pair to be viewed and/or manipulated (Figure 5). The same set of operations is available (*Add*, *Update*, *Delete*, and *Find*).

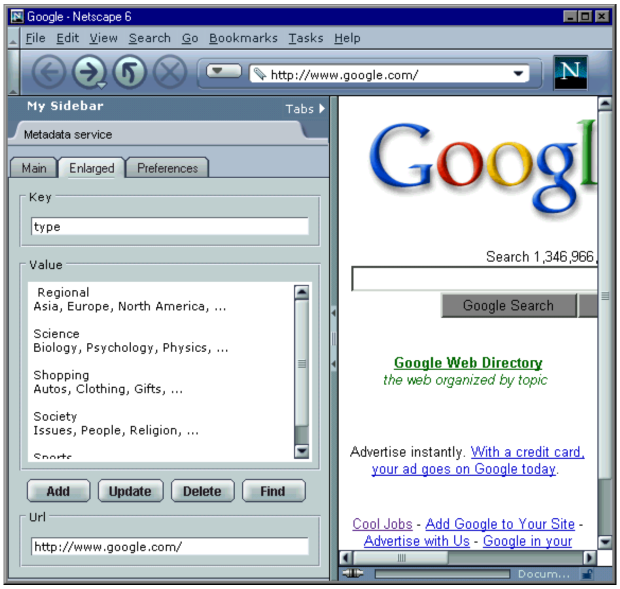

Figure 5. The enlarged view of the Construct metadata service sidebar in Netscape 6.

The *preferences* view allows users to choose which metadata service to use (Figure 6).

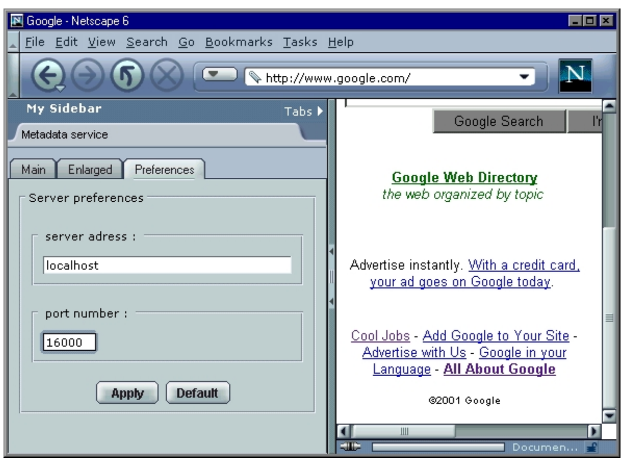

Figure 6. The preferences view of the Construct metadata service sidebar in Netscape 6.

Users of Netscape 6 can easily work with metadata when the metadata service sidebar is displayed. Whenever a new document is loaded (based on its URL), the main view in the sidebar displays automatically any available metadata for the document associated with the URL (if *auto update* is selected). To add a new metadata key, the user enters the name of the new key in the *key* field and presses the *Add* button. Then, the sidebar will switch to the enlarged view where the user enters the value of the key. Key values can be updated by selecting the key in the main view and pushing the *Update* button. Then, the value can be updated in the enlarged view. Finally, keys can be deleted by selecting them in the main view and pushing the *Delete* button.

As illustrated above, the metadata service is able to support all the operations on metadata in the scenario in Section 2.4.

3.2.3 Spatial structure service: Spatial hypermedia technology is designed to facilitate information analysis [19], which is an important task in many types of scholarly work processes. Spatial hypermedia applications help users to bring order to an information space. Typically, these systems provide a graphical representation of the information space in which the information objects being analyzed correspond to objects on a screen. By arranging and manipulating the object representations visually, users are able to identify similarities between objects, group related objects together, and identify repeated patterns or structures among the objects in the information space. Examples of spatial hypermedia systems include VIKI [19] and CAOS [25].

A spatial tool is currently under development in the Construct project. As a starting point (see Figure 7), the *Construct space tool* provides an information space and the ability to create objects (boxes) in the space and relations between objects (connections).

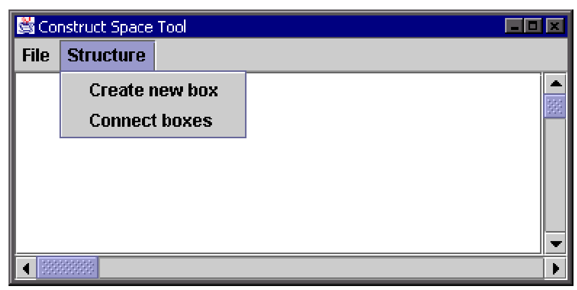

Figure 7. The Construct space tool.

The space tool supports different types of relations between objects within a space. The informal kinds are illustrated by co-locating objects in the information space (indicating that the objects somehow belong together). The more formal kinds of relationships can be visualized by creating lines (connections) between the objects (boxes).

In Figure 8, the user of the space tool has created four objects (*open hypermedia*, *issues*, *people*, and *systems*). The user has also decided that there are strong (formal) relationships between the objects. The *open hypermedia* object has been placed above the *issues*, *people*, and *systems* objects indicating a parent-child type of ordering between the objects.

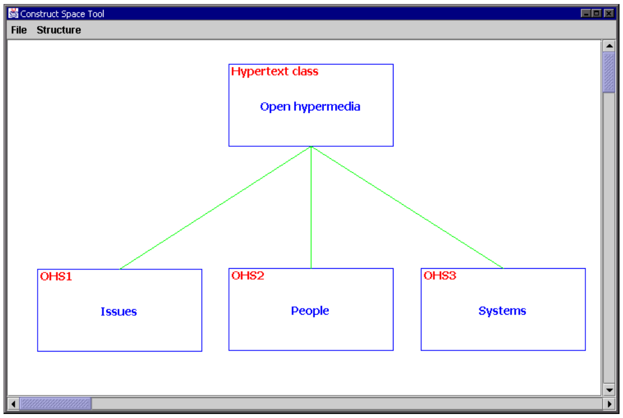

Figure 8. A parent-child type of ordering between objects in the space tool.

The space tool supports a hierarchical ordering of the information space. Objects in the space tool can include a subspace inside them. This is indicated in Figure 9 where the *issues*, *people* and *systems* objects each have been renamed (prefixed with "OHS") and transformed into their own information space (indicated by the text in the box: *Double click to open*).

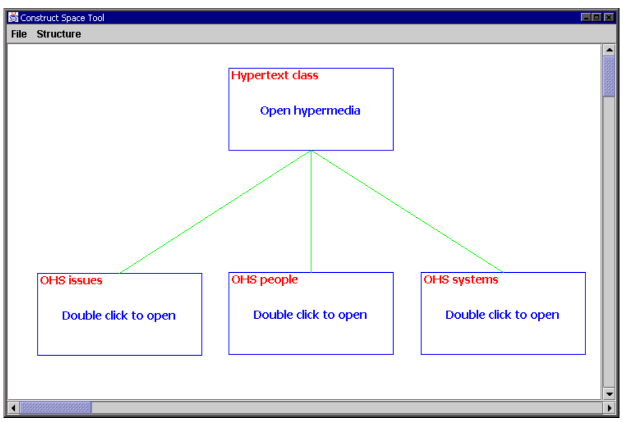

Figure 9. The space tool supports a hierarchical ordering of the information space.

Figure 10 shows the information space inside the *OHS issues* object. The space displays twelve objects each representing an open hypermedia research issue. At this point, the user has managed to group some of the objects into clusters of related issues by placing them next to each other. For instance, the user has indicated that the issues *Foundation services*, *Access control*, and *Version control* belong together. A next step in the information analysis process could be to create a subspace inside the *Foundation services* object and add additional related issues to the space (e.g., notification control, concurrency control, and structure storage).

The four objects (issues) to the right of the screen are not yet related to other objects (issues).

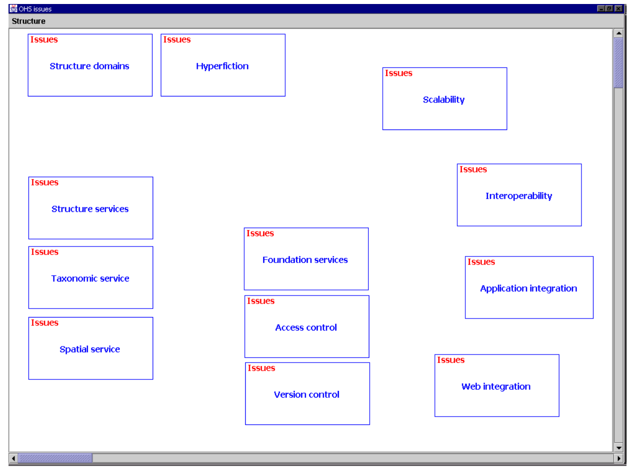

Figure 10. The information space inside the OHS issues object.

As illustrated above, the space tool can support the information analysis tasks described in the scenario in Section 2.4. Inspired by other spatial tools, we plan to enhance the Construct space tool with spatial parsing and better support for colors of boxes and appearance of boxes [19, 25]. As an example, this will allow the user to make related objects use the same background color to better distinguish them from other (unrelated) objects.

3.2.4 Taxonomic structure service: Taxonomic hypermedia systems are designed to support classification work [21, 24], which is another task involved in scholarly work. Classification work is concerned with organizing material in formal, rigid structures displaying a hierarchical relationship among the objects. Taxonomic tools have many of the same requirements as spatial tools. The spatial tool with its informal structuring features is typically used early in the information analysis process when the scholar is trying to make sense of the material at hand. The taxonomic tool with its more defined, formal structuring features is typically used later in the information analysis process when the scholar has a much better understanding of the material and is ready to start classifying it.

A taxonomic tool is currently under development. The first version of the *Construct tax tool* is shown in Figure 11. The tax tool provides three primary structural abstractions: *taxonomy*, *taxon*, and *specimen*. A taxonomy is an entire classification structure (tree). Taxons are inner nodes and specimens are leafs in the classification structure.

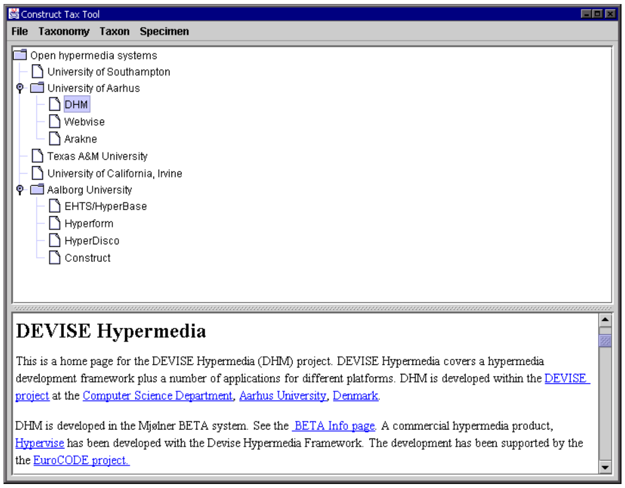

Figure 11. The Construct tax tool.

In the upper part of the tax tool (Figure 11), the user has organized the open hypermedia systems objects in a classification structure. The root taxon is named open hypermedia systems. The next level in the tree contains taxons representing research institutions developing open hypermedia systems. Two of these taxons (inner nodes) have been expanded to show the specimens (leafs). Each specimen represents an open hypermedia system that has been developed.

Both taxons and specimens can have files attached to them. The current selected leaf (DHM) has an HTML file attached to it that describes the Devise Hypermedia project at Aarhus University [10]. The lower part of the tax tool can display HTML files.

As illustrated, the tax tool can support classification work, which was also part of the scenario in Section 2.4. In future versions, the tax tool will be enhanced with better support for reorganization and versioning of the classification structures.

3.2.5 Data mining services: The data mining service is still under development. The first prototype of the data mining service and a data mining tool is expected to be operational during the Summer of 2001. This version will support all the data mining operations mentioned in the scenario in Section 2.4.

3.2.6 Cooperation services: The cooperation services are also still under development. The cooperation services consist of three parts: a service to manage shared sessions, a service and a tool to support awareness among multiple scholars (loosely coupled mode), and a formal cooperation service supporting synchronous collaborative work (tightly coupled mode). The first two types of collaboration services are expected to be operational in the Fall of 2001. The formal cooperation service is expected to be ready early in 2002.

4 Future work

The planned future work in the Construct project falls into several categories (in addition to the work in progress described in Section 3.2).

Defining new digital library settings. We are currently deploying our tools and services within the SIGMOD digital library accessible over our local area network. A natural next step is to deploy the tools and services to documents on the WWW. This can be seen as an attempt to scale up the environment to provide collaborative services on the Internet. We also have plans to "scale down" the environment to run on a single computer. The size of local hard disks continues to grow rapidly. In a few years, personal computers will have 100s of GB of local storage containing large amounts of personal information such as photographs, video clips, documents, etc., and then users will need tools to structure all their personal data.

Enhancing tools and services. We plan to enhance the functionality of existing tools and services. For instance, the set of data mining algorithms is open-ended. Many new technologies and algorithms as well as variations of existing technologies and algorithms can be applied to this problem domain.

Building tools and services. New tools and services providing support for additional work practices will be incorporated into the environment – for instance argumentation support (e.g., [4, 26]) and workflow services (e.g., [28]).

Building variations of tools and services. Variations of existing services will be built such as a new metadata service based on the Dublin Core Metadata Initiative [6]. The Dublin Core has 15 predefined (default) key/value pairs dealing with *content*, *intellectual property*, and *instantiation*.

5 Conclusions

The development of tools and services for knowledge discovery, management, and structuring in digital libraries in the context of the Construct project has come a long way. In Section 2, we presented an advanced scholarly work scenario involving several scholarly work processes (e.g., information analysis and classification) and several Construct tools and services. In Section 3, we showed that we have many of the tools and services in place necessary to support the presented scenario.

Future work will enhance the environment in many ways. The work practices described in the scenario will be available on the Internet to multiple collaborating scholars and additional work practices will be supported (e.g., argumentation support).

Acknowledgments

Many people have been involved in the development of Construct tools and services. We wish to acknowledge the following people for their contributions to the Construct project: our colleagues at Aalborg University Esbjerg Peter J. Nürnberg, Dragos Arotaritei, and Samir Tata, our close collaborators Sigi Reich, Salzburg Research, Salzburg, Austria and Jörg Haake, GMD-IPSI, Darmstadt, Germany, and the visiting Socrates students from the University of Brest, Bretagne, France, Jérôme Fahler, Stèphane Blondin, Yann Neveu, Youenn Guervilly, and Yann Le Doaré.

References

1. Anderson, K., Taylor, R., and Whitehead, E. J. 1994. Chimera: Hypertext for heterogeneous software environments. In *Proceedings of the 1994 ACM European Conference on Hypertext*, (Edinburgh, Scotland, Sep.), 94-107.
2. Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., and Secret, A. 1994. The World Wide Web. *Communications of the ACM, 37*(8), 76-82.
3. Bush, V. 1945. As we may think. *Atlantic Monthly 176*(1*)*, 101-108.
4. Conklin, J. E., and Begeman, M. L. 1988. gIBIS: A hypertext tool for exploratory policy discussion. In *Proceedings of the 1988 ACM Conference on Computer Supported Cooperative Work*, (Portland, OR, Sep.), 76-82.
5. Construct. 2001. http://www.cs.aue.auc.dk/construct.
6. Dublin Core. 2001. http://www.dublincore.org.
7. Engelbart, D. C. 1962. Augmenting human intellect: A conceptual framework. Stanford Research Institute Technical Report AFOSR-3223, Contract AF 49(638)-1024, Palo Alto, CA.
8. Fox, E. A., Akscyn, R. M., Furuta, R. K., and Leggett, J. J. 1995. Introduction to special issue on digital libraries. *Communication of the ACM 38*(4), 23-28.
9. Graham, P. 1995. The digital research library: Tasks and commitments. In *Proceedings of the 1995 ACM Digital Libraries Conference*, (Austin, TX, Jun.), 49-56.
10. Grønbæk, K. 2001. The Devise Hypermedia Project and Systems. http://www.daimi.au.dk/~kgronbak/DHM/DHMHome.html.
11. Grønbæk, K., and Trigg, R. 1999. From Web to Workplace – Designing Open Hypermedia Systems. MIT Press.
12. Hall, W., Davis, H., and Hutchings, G. 1996. *Rethinking Hypermedia – The Microcosm Approach.* Kluwer Academic Publishers.
13. Hicks, D. L., Tochtermann, K., Kussmaul, A. 2000. Augmenting digital catalogue functionality with support for customization. In *Proceedings of the Third International Conference on Asian Digital Libraries*, (Seoul, Korea, Dec.), 155-161.
14. Hicks, D. L., Tochtermann, K., Rose, T., Eich, S. 1999. Using metadata to support customization in digital libraries. In *Proceedings of the Third IEEE Meta-Data Conference*, (Bethesda, MD, Apr.).
15. Java Technology. 2001. http://www.javasoft.com.
16. Ley, M. Ed. 2001. ACM SIGMOD Anthology. http://www.informatik.uni-trier.de/~ley/db/anthology.html.
17. Ley, M. Ed. 2001. Digital Bibliography and Library Project. http://dblp.uni-trier.de.
18. Kacmar, C., Ed. 1995. Special issue on digital libraries. *SIGLINK Newsletter 4*(2), 2-28.
19. Marshall, C. C., and Shipman, F. M. 1995. Spatial hypertext: Designing for change. *Communications of the ACM, 38*(8), 88-97.
20. Nürnberg, P. J., Wiil, U. K., and Leggett, J. J. 1998. Structuring facilities in digital libraries. In *Proceedings of the Second European Conference on Digital Libraries*, (Crete, Greece, Sep.), 295-313.
21. Nürnberg, P. J., Schneider, E. R., and Leggett, J. J. 1996. Designing digital libraries for the hyper-literate age. *Journal of Universal Computer Science 2*(9).
22. Nürnberg, P. J., Leggett, J. J., Schneider, E., R., and Schnase, J. L. 1996. HOSS: A new paradigm for computing. In *Proceedings of the 1996 ACM Hypertext Conference*, (Washington, DC, Mar.), 194-202.
23. Nürnberg, P. J., Furuta, R. K., Leggett, J. J., Marshall, C. C., and Shipman, F. M. 1995. Digital libraries: Issues and architectures. In *Proceedings of the 1995 ACM Digital Libraries Conference*, (Austin, TX, Jun.), 147-153.
24. Parunak, H. 1991. Don't link me in: Set based hypermedia for taxonomic reasoning. In *Proceedings of the 1991 ACM Hypertext Conference*, (San Antonio, TX, Dec.), 233-242.
25. Reinert, O., Bucka-Lassen, D., Pedersen, C. A., and Nürnberg, P. J. 1999. CAOS: A collaborative and open spatial structure service component with incremental spatial parsing. In *Proceedings of 1999 ACM Hypertext Conference*, (Darmstadt, Germany, Feb.), 49-50.
26. Smolensky, P., Bell, B., Fox, B., King, R., and Lewis, C. 1987. Constraint-based hypertext for argumentation. In *Proceedings of the 1987 ACM Hypertext Conference*, (Chapel Hill, NC, Nov.), 215-245.
27. Stotts, D., Smith, J., Dewan, P., Jeffay, K., Smith, F. D., Weiss, S., Coggins, J., and Oliver, W. 1994. A patterned injury digital library for collaborative forensic medicine. In *Proceedings of the 1994 ACM Digital Libraries Conference*, (College Station, TX, Jun.), 25-33.
28. Wang, W., and Haake, J. M. 1999. Supporting workflow using the open hypermedia approach. In *Proceedings of the First Workshop on Structural Computing*, Technical Report AUE-CS-99-04, Aalborg University Esbjerg, Denmark, 12-17.
29. Wessner, M., Beck-Wilson, J., and Pfister, H. H. 1998. CLear – a cooperative distributed learning environment. In *Proceedings of the 1998 ED-MEDIA/ED-TELECOM Conference*, (Freiburg, Germany, Jun.), 1876-1877.

30. Wiil, U. K., Hicks, D. L., and Nürnberg, P. J. 2001. Multiple open services: A new approach to service provision in open hypermedia systems. In *Proceedings of the 2001 ACM Hypertext Conference*, (Århus, Denmark, Aug.).
31. Wiil, U. K., and Hicks D. L. 2000. HyTech: Development of hypermedia technology for digital libraries supporting scholarly work. In *Proceedings of the International Workshop on Cooperative Internet Computing* (CIC 2000), (Hong Kong, China, Nov.), 64-71.
32. Wiil, U. K., Nürnberg, P. J., Hicks, D. L., and Reich, S. 2000. A development environment for building component-based open hypermedia systems. In *Proceedings of the 2000 ACM Hypertext Conference*, (San Antonio, TX, Jun.), 266-267.
33. Wiil, U. K., and Nürnberg, P. J. 1999. Evolving hypermedia middleware services: Lessons and observations. In *Proceedings of the 1999 ACM Symposium on Applied Computing*, (San Antonio, TX, Feb.), 427-436.
34. Wiil, U. K., and Leggett, J. J. 1997. Workspaces: The HyperDisco approach to Internet distribution. In *Proceedings of the 1997 ACM Hypertext Conference*, (Southampton, UK, Apr.), 13-23.