Journal of Universal Computer Science, vol. 10, no. 9 (2004), 1110-1122
submitted: 29/4/04, accepted: 23/8/04, appeared: 28/9/04 © J.UCS

What we Expect from Digital Libraries

Heinz Dreher[1]
(Institute for Information Systems and Computer Media (IICM),
Graz University of Technology, Graz, Austria
hdreher@iicm.edu)

Harald Krottmaier
(Institute for Information Systems and Computer Media (IICM),
Graz University of Technology, Graz, Austria
hkrott@iicm.edu)

Hermann Maurer
(Institute for Information Systems and Computer Media (IICM),
Graz University of Technology, Graz, Austria
hmaurer@iicm.edu)

Abstract: Digital Libraries have been the subject of more than a decade of attention by researchers and developers, and yet in all this time the implementations have not matched the promises. By far the majority of systems have concentrated on content and provided limited or basic functions for users. In this article we offer a new look at what can be expected from a digital library system based on contemporary developments in Information and Communications Systems and Technology. First, we sketch out the basic functions which are provided to support finding and accessing material by a reader. Next we explain some extended functions which support the use and re-use of documents - links and annotations – and the need to support learners in addition to readers and writers. Finally, we present our visions for a modern digital library and e-Learning portal system which includes for example intelligent and conceptual search support including results visualization, white lists, and adaptive user interfaces.

Key Words: active document; annotation; black list; conceptual search; digital library; e-Learning; Hyperwave; links to the future; portal-systems; repositories of knowledge; searching; visualization; white list.
Categories: H.3, H.4, H.5

1 Basic Functions of Digital Libraries

Digital libraries are now widely available and implemented on different information or document management systems. Many of them share some basic functions which are discussed in the following section. Some functions which should be available to the users but are rarely implemented are also discussed.

Material stored in digital libraries is usually organized in some way. Different types of index-pages are usually generated automatically and presented to the user for

[1] On sabbatical from Curtin University of Technology, Perth, Western Australia.

browsing through the information-space. Articles by author, articles published in an issue or volume, articles by some category, to name just a few different types of indices. The ACM-computing classification schema is often used in computer-science related digital libraries.

Users access material differently depending on their aims. They either browse through the material stored on the server-side, or search for a variety of terms in different types of information: search by author, by title, by keywords etc. is offered to users by most systems. To narrow the search scope many systems support a kind of *search in this area-function*. Modern systems work with a variety of knowledge-management tools to provide some flexibility and tolerance in keyword construction, for example providing typographical error correction or review.

It is not possible for a single library to store every single document about a specific area. Thus resources available in external systems (i.e. other digital libraries) must be integrated and reused. A system should provide users with access to content regardless of storage-location. Document content as well as services provided by the remote digital library should be available to users. Unfortunately this portal-approach is not commonly used in digital libraries. Users are required to explicitly open another web-browser, connect to the selected digital library and repeat the search-query on the new system. This is ridiculous nowadays. The portal-software should be responsible for distributing search-queries to all selected digital libraries and provide search-results to users. This list of results must be presented to the user in a usable and re-usable fashion, permitting sorting, emailing, saving and the like. Users are (usually) unconcerned with the particular details of the digital library hosting an article, the article itself being the primary object of interest to the user!

Using a portal-approach for a digital library implicitly leads to many problems, including access rights to content, access to services provided by the remote system, selection of remote systems, handing of the results of functions called at the remote system to mention just the most obvious ones.

Access to services provided by digital library is often implemented using wrapper-software (e.g. Lixto, [Baumgartner et al. 2001]). These tools are usually parsers, using regular expressions to access HTML documents (or any other format) and to extract the relevant parts of a result-document. Depending on the wrapper-library used, this approach may fail if the design of the remote-server changes. Using XML-based remote service invocation (for example using SOAP-calls, simple object access protocol, [SOAP Primer 2003]) will make it easier for the portal-software developer to access remote services.

Many digital libraries and learning environments adapt the interface to the user. Some of the systems also implement personalization, i.e. adaptation of the content, personalized listings of content, recommendations to content of interest to the user etc. At the moment, users have to express their interest in certain topics via different user interfaces for every system. If interests change, users must change their user-profiles on every system. This task is as tedious and ridiculous as the manual distribution of search-queries discussed above. A common understanding of user-profiles is therefore absolutely essential. Portal-systems must provide this kind of adaptation to user interests and profiles.

Single user interests may be extended to interests of students in a specific lecture or research group. In an e-Learning environment the reading-material relating to a

certain topic may consist of several resources held by different digital libraries and various e-Learning systems or other web-based systems. This selection of course-material should be automatically gathered together and the resulting collection presented to all members of the course.

Another very powerful service of a digital library is a push-service, known as alerting service. Many journals use different alert mechanisms to notify registered users. For example, a user may be interested in all newly arrived articles, or articles matching some query in the full-text, abstract, category/classification, or title, or in new articles written by some particular author. Unfortunately, such a feature is presented in different ways by differing systems.

Hermes, a system described in [Faensen 2001], is a solution to this problem. This service filters notifications of published articles before they are sent to the user. A user profile -- like email address, search queries for title and/or abstract -- must be defined by the user just once and can then be used in many different systems.

This section discussed some common features such as browsing, searching, integrating different sources of information, personalization and alerting services. A modern digital library must implement all these features.

2 Extended Functions of Digital Libraries

In this section we take a look beyond the basic functionality already implemented in many digital libraries. First we take a closer look at the idea of annotations and consider some implications. Thereafter we explore links, created automatically or manually, by users or a group of users.

Annotations are basically notes attached to documents (e.g. [Andrews et al. 1994]). First implementations supported just text-based notes. However, since annotations are simple objects in a database, it is possible to attach arbitrary multi-media documents as annotations when using a Hyperwave Information Server (HWIS, [Hyperwave 2004]) as database.

In Hyperwave links are stored in a link-database. This implies that it is possible to add links to documents which do not support links per se. To give an example: PostScript was designed to be printed on high quality printers, not to be viewed online. Therefore Adobe did not take hyperlinks into consideration when specifying the document format. However, in Hyperwave it is possible to add links even to PostScript documents, because links are stored in the link-database rather than the document itself. Obviously, it is necessary to use a specially designed viewer to present the links to the user.

PostScript is now superseded by Adobe's Portable Document Format (PDF) where links are specified and viewers are available for many operating systems. Features of PDF include annotations, minor editing, highlighting etc. To create links in PDF it is necessary to use specially designed programs such as Adobe's Acrobat. Unfortunately this program is not available for every client platform, as an example, clients on Unix-platforms are precluded from the editing process.

Annotations are available in J.UCS [J.UCS 2004] since 1994. It is possible to add annotations to objects stored in the database. These objects are then linked via typed-links (link-type is *annotation*) to the source object. To add even more information to

links (such as: this annotation is a question, or answer, note etc.) we added additional attributes to the link object.

With these additional attributes it is possible to create *Active Annotations* [Heinrich and Maurer 2000] and thus facilitate discussion about articles. The corresponding author of an article is alerted by the system via an email about the newly created annotation, making it possible to answer questions or explain some thoughts in more detail. The publishing process therefore does not end after a successful submission of an article. It continues as long as questions are asked or comments are made.

Questions may be answered by experienced users, or by the author as in the above example, but any other expert in the field may answer a question. Questions may also be answered by the system! This concept is called *Active Documents* and was introduced in [Heinrich and Maurer 2000]. It had been shown, that different users ask similar questions. If there is an answer to a similar question, than this particular answer may also be an answer to the asked question. Modern knowledge management systems are able to measure similarity between text documents, thus similar questions can be found and a relevant answer can be presented to the user. If the user is not satisfied with the shown answers, the question may be forwarded to an expert.

Additional attributes of annotations are related to access-control (read, write, modify) of those annotations. Three types of annotations should be possible: private-, group- and public- annotations. Generally every user must be able to add at least private annotations to an article. These annotations are only visible to the user who created them. It must be possible for this user to change or delete the annotation.

Some users (e.g. users from a research-group) must be allowed to read and write group-annotations. This is necessary for group members who collaborate on a specific topic or task. A closed mini-discussion about an article with a defined group of users is therefore possible. As implemented in many discussion forums, it must not be possible to edit annotations after they have been annotated by another user for obvious reasons (stable annotations).

Public annotations are visible to every user; therefore the system and the administrator should take care of this feature. In some circumstances (e.g. in a learning environment) public annotations must be restricted. Readers should, for example, be forced to read certain parts of a document before being allowed to annotate an article or ask questions about an article. Online user-tracking and user activity-logging are evident pre-requests to implement such features.

To get rid of *spam-annotations*, i.e. annotations, which contain material unrelated to an article, annotations should also be integrated in a quality or relevance rating system. Users should be able to judge annotations as ‘great’ or ‘poor’, or according to some suitable set of criteria. Thus many ‘poor’ annotations will disappear from the listing of annotations and users can therefore be spared the time reading them [Krottmaier and Helic 2002].

Some users are skilled in reading a hyper-linked version of an article on screen; others want to print out the PostScript or PDF-formatted version of the article. Different document formats are therefore offered to the user, and in J.UCS for example, HTML, PostScript and PDF are the supported document-formats. The contribution of an author is coded in these different document-formats and is then stored in a so called article-collection. Abstracts are converted to HTML and are

available free to users. Annotations to an article are attached to the article-collection and are therefore available for viewing in all the different document-formats. It is possible (and already implemented in a prototype) to simply add a page of annotations to the PDF-formatted paper.

As mentioned at the beginning of this section, links must be managed separate to the content for various reasons. Therefore everyone who is allowed to read the content is also allowed to attach links to the content. It must be possible to create a personal collection of articles or a personal tour through a thematic collection, thus creating intricate associative trails envisioned by Vannevar Bush [Bush 1945].

Personalization is not just about content and interface adaptation, but also about links. Adding a link to some content is like adding an annotation to the content. When user-access-rights are attached to link-objects, it is possible to simply deactivate unrelated links. To give an example of a digital library used as background library in an e-Learning course: students new to a course may need more links from the content to a dictionary or thesaurus explaining a word as compared to more experienced students from later course stages.

In a digital journal hosting scientific articles many types of personalized links are possible. Beyond typical features such as table of contents, links to authors' homepages, email addresses etc. other intra- and extra-document links must support the reader while exploring an article.

A references section is typically included in an article, providing intra-document links from the citation in the text to the reference section and is mostly created by automatic link creation tools. Using a highly sophisticated server-system with bidirectional-link management allows the user also to retrace from the reference section to the corresponding location of the citation in the text. This is an important feature which facilitates use and reuse of the body of knowledge reposing in an article.

Extra-document links are links to other documents stored either on the same server system or stored on a remote server system. The 'surrounding' of a document (i.e. a cluster with similar documents) with documents containing similar concepts or ideas might be visualized via a knowledge map already introduced in many knowledge management systems. A knowledge map with similar articles is easy to create with intra-server documents because the full-text or an index is available to the system.

Using automatically generated links allows us to implement many novel features in a digital library environment. A simple but powerful idea is the idea of *links to the future*. Imagine the following situation: An article A cites an article B. B was published somewhere in the past using an arbitrary system. Currently, most systems create links from article A to article B. Since links in ordinary web-technology are unidirectional, no reader of article B will ever notice (without going to the considerable effort to consult citation indexes and run the risk of losing sight of the main task) that the author of article A cited this article. One can claim that it is possible to create automatically two links: from article A to article B and vice-versa. However, this is very often not possible because one needs write access at the system where article B was published to create a new link. Using a Hyperwave Information Server as publishing system will facilitate the implementation of this idea of links to the future. As mentioned above, a link-database is used in Hyperwave, i.e. links are

stored separately to the content. No write-access is necessary to create links to a document.

Using extra-server documents might be difficult because of access rights to the documents. To give an example: In J.UCS the content is indexed by the Google search engine via a special crawling agent (known as *googlebot*). Therefore search-queries initiated by the Google search engine are also applied to J.UCS-articles. If the query matches articles are shown in the listing. However, only users subscribed to J.UCS are able to access the full-text of the article.

Additionally there might be an implementation of links to traditional material. This traditional material (i.e. books or printed journals) is sometimes available at the local library. The local library usually offers a searchable catalog of available material. Sophisticated systems provide users with status information such as "the book is currently on loan by another user" or "the book is available". Why not reuse this information and integrate it in the listing of references? Many wrapper-tools are already available to access this kind of information from an online-catalog system. A red-light might indicate that the book is available in the local library, but currently borrowed by another user, a green-light might indicate that the book is available to the user. Via some special link it should be possible to reserve the printed book in the library. In case the book is not available in the library but many articles are citing it, the librarian may order the book. In certain applications the system should provide links to an online bookstore to make it easier for the user to acquire the referenced book.

3 e-Learning

Reports and case studies of e-Learning systems are prevalent in recent literature, and reflect the popularization of systems such as WebCT (http://www.webct.com/) and BlackBoard (http://www.blackboard.com/) among others. Visiting these sites reveals the overwhelming use being for course presentation and content storage/access purposes - that is, e-Learning systems are being used to support students in their learning activities. Projects such as the "Making of America" [Mead and Gay 1995](p11) and more recently the "Digital Library for Earth System Education" [Sumner and Dawe 2001](p416) are examples of the massive effort being spent on building 'content' or *repositories of knowledge* for use by students in their learning endeavors.

Just as traditional libraries support students' learning via the 'content' of the library, so should a digital library. However, there is much more to be done. Learning does not take place simply by accessing content. The information or knowledge must be processed by the learner's mind. Support can be provided in structuring, reflecting - analyzing and synthesizing the knowledge, as is evidenced by the numerous so-called Knowledge Management (KM) systems, among which Hyperwave (http://www.hyperwave.com) would surely be a leader, being adopted largely by the business sector.

Whilst our vision of a digital library obviously includes content, access to content, and support to 'process' that content, we see further e-Learning functions in need of development. For example, [Carlson and Larralde 1995] report on software to teach reading comprehension, in which they distinguish between "making meaning"

and "extracting meaning" (p156) – clearly there is recognition of the need to do more than simply provide access to content. [Dreher et al. 1994] developed their *Active Writing Project*, a hypertext writing environment to support children constructing their words and stories based on the real world models created with LEGO (http://www.lego.com) to support and empower learners in constructing their 'worlds' and producing a written accompanying story. Such examples of computer environments developed to support learning in the constructivist mode [Bruner 1966] may serve to guide the design of e-Learning functionality provided by digital library systems. [Maurer and Lennon 1995] provide an early version of what they boldly termed (and one may observe, with appropriate justification, for the powerful Hyper-G/Hyperwave system was available for their use) "ultimate teaching and learning tool" (p719) functionality in digital library systems.

Such systems as described by [Mead and Gay 1995] and [Sumner and Dawe 2001] provide us with a clear indication of the direction digital library research and development may take: there is a growing emphasis on supporting far richer learning paradigms than the mere *browse* or simple access of material. For example, [Scerbakov 2004] builds scenario-based learning function into his WBT-Master e-Learning environment in which much more than *access*, *browse*, and *discussion* is supported. A scenario consistent with the educational design of a course can be realized by students and may include reflection, assessment, and feedback. This is clearly a move in the direction of *teaching* support in addition to the usual *learning* support. [Helic et al. 2003] report on supporting "new and innovative teaching and learning paradigms" (p2189) in their reaction to the usual Web-based training systems which reflect available technology as opposed to implementing good teaching and learning theory. Strong emphasis on managing assignments is reported in the design for the DocML digital library of university data being developed in Greece [Papadakis et al. 2002], and [Lennon and Maurer 2003] explain that support for *active documents* is presently realizable in the e-Learning suite of Hyperwave. The concept of *active documents* is highly desirable in facilitating students to ask questions of the 'document' (the document author actually) thereby provoking the publication of a question answer pair which may subsequently prove useful in other student-learning situations. This can be seen as some sort of automatic-teaching to support learning-on-demand.

A common thread running through the recent literature is the hope that systems will be developed which can provide a wider range of support for learning and teaching, systems that empower, and we are more than optimistic that with proven and reliable Knowledge Management Systems such as Hyperwave, together with careful consideration of the educational possibilities and imperatives, empowering digital library systems, which support e-Learning in the broad sense of the term, are imminent in their realization.

4 Visions

4.1 Intelligent Search

Prior to the advent of the Web, locating research materials in academic libraries was a laborious process indeed. Even then, with a relatively restricted search scope of 'the

library', possibly extending to some related libraries and using the 'inter-library loan' device, we were able to consult relevant material after many days or even weeks of labor. Now, with a dramatically increased search scope (we want to access everything in the known world), digital versions of the articles we require, and computers to carry out our search commands we often discover that relative to the bygone era we may not be so very far advanced as one might expect. In their "model of the research process using un-integrated systems and traditional methods", [Arant and Payne 2001](Fig 2, p68) point out that librarians act as intelligent support systems to assist users in searching for and locating the required material. Largely through the Web and Internet, the entire store of the world's knowledge is now our scope, and we desperately need intelligent search mechanisms, agents, interfaces and portals.

Many scholars have experienced that the typical search engines being used to locate information on the Web are not so effective in locating research documents and academic publications accessible via the Internet. There are numerous reasons for this: document format – PDF and Postscript files are often not 'indexed' by search engines; search scope – the usual scope is WWW documents and thus does not include scholarly databases or other repositories of published knowledge; idiosyncratic search parameter specification interfaces and search rules – the variety of search interfaces is so large as to create cognitive overload, and of course, markedly slow the discovery of relevant documents due to the need for tedious interactions. The above point is made abundantly clear in the 2003 UNESCO report: *Internet in Education* (p54):

> "Data, documents, books, libraries and bigger encyclopedias are available at home after a successful search. However the Internet is in danger of becoming a labyrinth for the users that have to deal with different interfaces, with the communication systems of the programmes, which elements inform them on what is possible to do or what is happening, so they can end up being confused by the differences that the diverse programmes establish for one operation."

He and Hui have proposed one solution to the *search results overload* problem which recognizes the importance of searching by authors who publish in similar research fields. Their PubSearch system "provides a document-clustering search and an author-clustering search based on the Web citation database" [He and Hui 2001](p274); [He and Hui 2002].

Major breakthroughs however are going to come once we realize the importance of supporting users' diverse mental models or cognitive maps of the knowledge domains being searched by them. [Mead and Gay 1995] listed among their Principles for digital library design "Hyperlink Tools" and "Annotation Tools" (pp13-14). [Dong and Agogino 2001] have also identified the need for "methods to organize the materials around personal context(s) rather than a prescriptive context." (p316). The recognition by digital library designers that individual users' idiosyncratic needs, knowledge or concept maps and search strategies can be supported will result in the cognitive empowerment being sought by scholars the world over. And whilst the literature has been mentioning such empowering 'devices' for more than a decade, there appears to be little implementation effort.

4.2 Visualisation of Search Results

One of the big issues relating to searching is the presentation results. Frequency tables, star ratings, and similar devices are now being used to augment the usual linear lists of search results. But more can be provided, as is seen in the DocML design [Papadakis et al. 2002] in which "Special attention is given to the visualization of the information repository's contents, which, in turn, facilitates more efficient acquisition and retrieval of stored information." (p379). [Mead and Gay 1995] mention that "concept maps" (p11) can be usefully deployed as part of the digital library interface.

[He and Hui 2001] explore a variety of methods to represent query results in meaningful ways, among which "Maps are provided to give a visual overview of the whole document collection with similar documents located close to each other" (p277). Repeated fine tuning of search query selection criteria could be contingent upon the graphical output indicating the attainment of a 'suitable' search result, say in terms of search term density per document, or number of documents, and so on.

We can take this idea one step further and envisage a search system capable of having the search criteria specified in the normal way, and in addition we would specify desirable search result parameters in the form of a visual device, target graph or map. Some special tools need to be built to support such interactions as that the *what-if* analysis tools provided for example in Microsoft Excel.

4.3 Conceptual Searching

When we are looking for something we form a concept of the target, the thing we are seeking, and then by some means locate and evaluate the things we have found for their suitability to the purpose. Typically we search textual repositories, and use words as the keys to match in the target. So called advanced search functions may permit an adjustment of scope, facilitate the use of attribute-fields (e.g. time, authorship, subject domain, and including the prevalent meta-data tags for WWW material, etc.), provide word variant, phonetic, or even multi-lingual support. All these devices are deployed in an effort to narrow down the search, and supposedly arriving at a manageable and useful set of search results. However, words in vacuo, are a poor substitute for the real thing, the concept. A variety of words can be used to form a concept, and if we want to find concepts, as opposed to words, we must rethink our search strategy.

When our word (key-word) searches produce a manageable quantity of results we may be content with the approximation which these key-words deliver in respect of our concepts. The intellect is then deployed to sift through the candidates in our hit-list, and relatively quickly the suitable targets can be delineated from the mass. Now the question must be, is our found material pertinent to the concept we had in mind? Such a system works for small to moderate quantities of knowledge, but as is more often the case now, we find ourselves with so much in the hit-list that it is unimaginable to even think about the next step, that of using the information we have found. Clearly help is needed.

In their article on digital library usability from a reuse perspective, [Sumner and Dawe 2001] report that "Faculty resist "cold" searching the web for a number of reasons: the thousands of hits returned by search engine queries and the time required to evaluate them; the frequently unrelated or low quality websites returned from a

search query; and the many links that are broken or no longer point to relevant material. In short, web searching is perceived as inefficient and frustrating because of the enormous amount of information to sort through and high variability of quality of results." (p420). Under such conditions one wonders why one began the search at all.

But quantity or hit-list reduction is not the only challenge we face. With key-word searching, or any other of the many approximations we create to represent the search concept, we have no way of knowing whether the search has delivered all (or most) of the potential targets. Systems using the *find more like this one* help in that respect, and facilitate an interaction between the searcher and that which is being searched. During this interaction, a concept may develop from that of the initial construction into something much more pertinent. [Dong and Agogino 2001] have mentioned the importance of organizing the information once it is found, and to that end suggest the use of "concept maps" (p317) among other information structuring devices.

We suggest that searching needs to provide for an interaction between the search concept and the search target, mediated by the searcher's intellect, with criteria and results being represented visually in addition to the usual methods. The search can be said to be successful when the searcher is satisfied.

4.4 White Lists

As already mentioned in the first chapter, it is impossible for a digital library to store every relevant document in the local system. We proposed a portal system to integrate material stored on any other system. However, questions still remain: Which systems should be integrated? Should there be a *White List*, or a *Black List* of systems? Should there be a *generic list*, or should lists be topic specific?

It had been shown, that the concept of *Black Lists*, i.e. lists with names or addresses of systems which contents should not be integrated in the portal, is very difficult to manage. Updates of these types of lists must be performed on regular basis. One must not mention that this is an expensive task. Most *Spam-Lists* work in this way.

In a digital library the concept of *White Lists*, i.e. names or addresses of systems, which should be integrated in search scopes or any other service, is much more appropriate. It had also been shown in learning environments, that this approach is easier to handle for teachers. If they know, that interesting material is available at server X, they simply suggest, that the systems should also perform operations with material stored there. Operations in this context are not limited to search operations. One may think also of annotation services, reference analysis, author lists, to mention just a few.

Specific *White Lists* for some readers are not necessarily optimal lists for other readers. Different kinds of lists must exist for different types of readers (i.e. groups of readers) and for specific topics. To give an example: details of a topic covered in a course for first grade students are different to those discussed in an elementary school. Therefore different external resources must be integrated for those different types of students and therefore different *White Lists* must exist.

4.5 Adaptive User Interface

User interfaces are needed to provide system functionality to the person wanting to invoke selected functions and should simplify the user's task. As system functionality increases, so usually does the interface complexity which can in turn lead to bewilderment or non-usability in extreme cases. There are a variety of techniques to deal with such situations. One is to provide say three different interfaces – a normal function version, a *lite* version, presenting only the most used functions, and perhaps a full function version. It is important to adhere to a design consistency as users may want to move between various versions as their experience grows and their needs change.

The idea of providing software versions to suit a particular user is not new and is seen at installation time of most modern software products, but selecting user interface options is not so prevalent. A well known example of user interface selection is PKZiP software (www.pkware.com) in which there exits a "classic" or "Web" version. One could imagine that choice to be automated dependent on the user context – if a Web browser was being used it may make sense to automatically offer the corresponding PKZiP interface.

Software systems can be built to be sensitive to user context, and this is indeed done by most e-commerce sites inter alia through the deployment of Cookies, but usually, or frequently, for purposes other than user interface adaptation. Good e-commerce sites will respond to basic user platforms attributes such as browser type/version, screen resolution, and helper application availability, but the adaptation is rarely dynamic.

Truly adaptive user interfaces learn from the experience of the user. There is clearly a need for interaction between the user-interface and a user-interface monitoring agent and also between the user and that agent. Users should be able to configure their own interfaces which initially would inherit the design of one of say, three main interface options mentioned earlier. As usage history is developed the interface should, optionally, adapt to suit that usage better, and the user is asked if a certain change, ascertained from usage/behavior analysis, should be implemented. The auto-suggestion of certain interface adjustments based on user behavior analysis should be sensible and not invasive or persistent and annoying.

There are some spin-off advantages of adaptive user interface agents one of which would be to provide what can be termed session control – much of the work done at computers cannot be completed in one sitting and therefore a system to preserve the workspace or workstage and facilitate a subsequent restart and continuation is directly implementable with the data monitored by such software.

5 Conclusions

In this article we have followed the digital library research and development ideas as presented in the literature and in existing digital libraries and observed the gap between design promises and implementation functionality. Consistent with contemporary Information and Communications Systems and Technology we have presented our expectations of a digital library system and detailed some of the empowering functions to be implemented.

Our ongoing research and development in digital library and e-Learning portal-systems is expected to result in detailed specifications within six months and implementation of the first phase within 18 months.

References

[Andrews et al. 1994] Andrews, K., Kappe, F., Maurer H. and Schmaranz, K. (1994): "On Second Generation Hypermedia Systems", *Journal of Universal Computer Science*, Vol. 0, Issue 0, pp. 127-136.

[Arant and Payne 2001] Arant, W. and Payne, L. (2001): "The common user interface in academic libraries: myth or reality", *Library Hi Tech*, Vol. 19, Nr. 1, pp. 63-76.

[Arms, 2000] Arms, W.Y. (2000): "Digital Libraries", MIT-Press

[Baumgartner et al. 2001] Baumgartner, R.; Flesca, S. and Gottlob, G. (2001): "Visual Web Information Extraction with Lixto", *VLDB Journal*, pp. 119-128.

[Bruner 1966] Bruner, J. (1966, 1986, 1990)
Accessed on 25/3/2004 http://www.artsined.com/teachingarts/Pedag/Constructivist.html

[Bush 1945] Bush, V. (1945): "As We May Think", *Atlantic Monthly*, July. Accessed on 25/2/2004 http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm

[Carlson and Larralde 1995] Carlson, P.A. and Larralde, V. (1995): "Combining Concept Mapping and Adaptive Advice to Teach Reading Comprehension", *Journal of Universal Computer Science* Vol. 1, Issue 3, pp. 156–161.

[DeMulder et al. 1993] DeMulder, R.V., van den Hoven, M.J., and Wildemast, C. (1993): "The concept of concept in 'conceptual' legal information retrieval", *9th BILETA Conference Building Systems*, 1st and 2nd April, John Moores University, Liverpool, UK.

[Dong and Agogino 2001] Dong, A., and Agogino, A.M. (2001): "Design Principles for the Information Architecture of a SMET Education Digital Library", *ACM+IEEE Joint Conference on Digital Libraries 2001*, June 24-28, Roanoke, Virginia, USA. ACM 1-58113-345--/01/0006

[Dreher et al. 1994] Dreher, H.V., Dreher, L.H., and McKaw, K. (1994): "The Active Writing Project - small movements in the real world", *Proceedings of APITITE 94 (Asia Pacific Information Technology in Training and Education) Conference*, Brisbane, Australia, June 28-July 2.

[Faensen et al. 2001] Faensen, D., Faulstich, L., Schweppe, H., Hinze, A. and Steidinger, A. (2001): "Hermes - A Notification Service for Digital Libraries", *Proceedings of First ACM/IEEE-CS Joint Conference on Digital Libraries*, Roanoke Virginia, USA.

[He and Hui 2001] He, Yulan and Hui, Siu Cheng (2001): "PubSearch: a Web citation-based retrieval system", *Library Hi Tech*, Vol. 19, Nr. 3, pp. 274-285.

[He and Hui 2002] He, Yulan and Hui, Siu Cheng (2002): "Mining a Web Citation Database for author co-citation analysis", *Information Processing & Management*, Vol. 38, Issue 4, July 2002, pp. 491-508. doi:10.1016/S0306-4573(01)00046-2

[Heinrich and Maurer 2000] Heinrich, E. and Maurer, H. (2000): "Active Documents: Concept, Implementation and Applications", *Journal of Universal Computer Science* Vol. 6, Issue 12, pp. 1197-1202.

[Helic et al. 2003] Helic, D.; Krottmaier, H.; Maurer, H. and Scerbakov, N. (2003): "Implementing Project-Based Learning in WBT Systems", *Proceedings of E-Learn 2003*, pp. 2189-2196, AACE, Charlottesville, USA.

[Hyperwave 2004] *Hyperwave Information Server*, http://www.hyperwave.com

[J.UCS 2004] *Journal of Universal Computer Science*, http://www.jucs.org

[Krottmaier and Helic 2002] Krottmaier, H. and Helic, D. (2002): "More than Passive Reading: Interactive Features in Digital Libraries", *Proceedings of E-Learn 2002*, pp 1734-1737, AACE, Charlottesville, USA, 2002.

[Lennon and Maurer 2003] Lennon, J., and Maurer, H. (2003): "Why it is Difficult to Introduce e-Learning into Schools And Some New Solutions", *Journal of Universal Computer Science* Vol. 9, Issue 10, pp. 1244-1257.

[Maurer and Lennon 1995] Maurer, H.., and Lennon, J. (1995): "Digital Libraries as Learning and Teaching Support", *Journal of Universal Computer Science* Vol. 1, Issue 11, pp. 719-727.

[Mead and Gay 1995] Mead, June P. and Gay, Geri (1995): "Concept Mapping: An Innovative Approach to Digital Library Design and Evaluation", *SIGOIS Bulletin*, Vol. 16, No.2, December, pp. 10-14.

[Papadakis et al. 2002] Papadakis, Ioannis; Karakoidas, Vassileios and Chrissikopoulos, Vassileios (2002): "DocML: a digital library of university data", *Library Hi Tech,* Vol. 20. Nr. 3. pp. 378±387 # MCB UP Limited . ISSN 0737-8831 DOI 10.1108/07378830210444577

[Scerbakov 2004] Scerbakov, N. (2004): in personal communications regarding "learning scenarios", March, IICM, TU-Graz (http://www.iicm.edu)

[Stern, 1999] Stern, D. (1999): "Digital Libraries Philosophies, Technical Design Considerations, and Example Scenarios", Haworth Press

[SOAP Primer 2003] "SOAP Version 1.2 Part 0: Primer" (2003): *W3C Recommendation*, Accessed on 25/2/2004 http://www.w3.org/TR/2003/REC-soap12-part0-20030624/

[Sumner and Dawe 2001] Sumner, T., and Dawe, M., (2001): "Design Principles for the Information Architecture of a SMET Education Digital Library", *ACM+IEEE Joint Conference on Digital Libraries 2001*, June 24-28, Roanoke, Virginia, USA. ACM 1-58113-345--/01/0006

[UNESCO *Internet in Education 2003*] UNESCO Institute for Information Technologies in Education, *Internet in Education -Support materials for educator*s, Moscow (2003) p. 54.