Electronic Publishing '01 –
2001 in the Digital Publishing Odyssey
A. Hübler et al. (Eds.)
IOS Press, 2001

Improving the Usability of a Digital Library

Harald Krottmaier
Institute for Information Processing and Computer Supported new Media
Inffeldgasse 16c, A-8010 Graz, Austria

Abstract

The Journal of Universal Computer Science (J.UCS) was introduced in 1994 and after more than 7 years of operation the service is still up and running. Articles published on the server are categorized in several ways (e.g. in the ACM Classification Scheme) to simplify browsing and finding articles again. Nevertheless users wants to create their own view on the material and their own categories. Traditionally a user creates some categories on the client-side in a bookmark-structure and stores the URLs of the according papers in this structure. But the use of bookmarks binds the user to a specific client on one single machine and so it is not possible to take full advantage of the restructured view. This is just one reason why it is helpful for the user to manage his personal view of the articles on the server-side. In this paper we introduce a personal workspace for registered users of the Journal of Universal Computer Science. It is possible to create and modify a personal view, add some personal and/or public comments of the articles and use the workspace as personal repository of published articles. We also show other advantages like personalized search scope, customization of the order of articles and some other useful features of the system.

Keywords: Hyperwave, Journal of Universal Computer Science, personalized content

1 Introduction

Typical digital libraries are static and passive systems. They do not adapt to users preferences, like colors and design of the interface, topic of interest, rearrangement of articles etc. Because every user is different in many ways, this issue must be considered in the design of an easy to use and helpful digital library. Passive in the context of digital libraries means that the user interacts with the system, but the system does not initiate an interaction with the user. This behavior can be omitted by sophisticated tools later described in this document.

There are several possibilities to address these two issues. To get rid of the static behavior, user preferences must be stored at some place. Due to the generic accessibility of one system, usually a standard web-browser is used to access and visualize the content digital library. Therefore preferences might be stored at the client-side, at a proxy-server or at the server-side.

Storing data at the client-side in some cookies (e.g. [Clarke, 2001]) is a problem when mobility, backup etc. is an issue. Just to address the problem of mobility: If one changes the client-computer due to failure or any other reason, the preferences are not accessible and therefore the server-system can't take these preferences into account when assembling the content of a page. Also if a user changes the type of browser on one single computer, the stored cookies for browser *a* are not accessible for browser *b*. These are just two reasons why it's not very reasonable to store personalized information.

A proxy-server is an intermediate server-system between client and server. If such a proxy is located at the intranet within some company, it will not be possible to access this intermediate server-system to get some user defined preferences. Therefore in some cases – e.g. for an exclusively company-wide digital library system this approach might be possible, but for an international, public available system only the following storage location of user preferences is proper: the server-side. All advantages (browser and client-computer independent access, central backup etc.) and disadvantages (failure of server implies unavailability of the service to all clients etc.) are immanent in this approach. But independent availability is a key-issue in our system [Maurer and Schmaranz, 1994] so the decision was to store all user defined preferences at the server-side. This paper is addressed to this storage location also know as workspace. Some requirements to the implementation of the workspace and some tools are described. These tools handle the problem of passivity of typical digital library systems. The following section describes the requirements for the workspace which will be implemented for the Journal of Universal Computer Science (J.UCS) to release the power of a digital library.

2 Requirements for a Workspace

As described above the workspace in our context is some storage place at the server. Another term for a workspace might be 'Homecollection' or 'Homepage' with some additional tools. Some parts might be accessible for all users or for some groups, other parts are private and therefore invisible for other users. Like in traditional libraries, there is a need for classification of the content. In J.UCS the categorization is done in several standard ways:

ACM Classification System: The first ACM classification system for the computing field was published in 1964. This system was developed several times to consider new topics in computer science [ACM, 1998] and makes it easy for a reader to access immediately the topic of interest without wasting time to find articles on a specific topic. For example, if one is interested in the topic of digital libraries, he will find related articles in category H (H...Information Systems), aspects of Information Interface and Presentation are found in category H.5, whereas the sub aspects of user interfaces in information systems are found in category H.5.2 User Interface.

By Time: Obviously newer articles are in some kind more interesting to users than old ones. There must be a simple distinction in the time-line of articles. Articles are published in monthly issues and are bound to a volume for each year. This makes it very easy for the user to recognize the time of publication of each article.

By Author: If one is interested in all papers published by one author, the related papers can be found simply by browsing to the corresponding collection. Due to the many articles already published in J.UCS, there is a need for further granulation of articles.

Please note that all these listings are generated on the fly and are therefore every time up to date. As a requirement for a private workspace, it must be possible to allow the users to create their own views on the material (i.e. articles, issues or volumes). If the user is interested in e.g. information systems (category H), why should the user see the articles of category F (Theory of Computation)?

To put that issue into practice users must define their topics of interest. This can be done in several ways: Either one user links the collection of category H to his assigned workspace, or the displaying routine for the category listing consider specific variables set by the user via some form. To take full advantage of the parent/children-relations of a Hyperwave Information Server (e.g. [Maurer, 1996], [Hyp, 2001]) it is clear, that the first approach (link some part – e.g. a paper or any other entity of the content – to the workspace) must be implemented. This can be either done automatically (via some cgi-script or some server-side Java Script) if the user fills out some form or manually by the user using some tools. If some user decides that his own categories should be available for other users, he simply assigns the appropriate rights to his own categories.

This approach immanently implements 'public bookmarks' and can be used in an background-library for some learning environment or web based training system. Note the big difference of that approach to the discussed possibilities described above. Unlike client-side bookmarks this workspace is not just available to other users (if the owner sets appropriate rights), but the structure is also available within *any* web-browser. Administrators are able to extract valuable information from the linking structure of the content and can thereby advise the editors about topics often viewed by users. A system explicitly implementing such a public/private bookmark-service is e.g. [MyB, 2001] but it is obvious, that a library system should implement this feature immanently to take full advantage of the system.

One can imagine that due to personal preferences and intention of some collection, the sort order of the linked entities may differ. To give an example: One may collect some articles in one collection *'History of Information Systems'* and therefore wants some sort order of the articles ascending in date of publication. Another user might create a collection with the same articles *'Latest Developments in Information Systems'*, but with another sort order (descending in date of publication). It is obvious, that no physical copies are allowed but simple transclusions (reference links, see e.g. [Nelson, 1987]) must be implemented.

If another arbitrary sort order of articles must be possible, context specific attributes to the articles must be implemented. This can either be achieved via specific container objects or via attributes in the parent collection of the linked articles. Another feature easily implemented with that technology will be a global history of what the user accessed on the server system.

One can see, that this kind of workspace has some powerful features and will make the life much easier for users of the library. The next step of implementation will allow to rearrange and collect mate-

rial (i.e. everything addressable by an Uniform Resource Identifier (URI, [Berners-Lee et al., 1998]) not stored on the local Hyperwave Information Server but anywhere at the web [Krottmaier and Maurer, 2001].

As pointed out above a personal workspace can improve the usability of the library enormously. But so far, only the navigation through the material of the library and finding articles again is improved in comparison to static libraries. In the following section some improvements in existing functions and new functions provided by the library are described.

3 More than just Reading...

As mentioned in section 1, a user should work with the library and not just read articles in it. It is obvious that a personal workspace on the library server is just the first step to a useful environment.

Traditional features for adding some kind of value to the articles are annotations or comments to articles. J.UCS was one of the first systems implementing that feature. Nowadays nearly every electronic library system supports annotations in some way. Annotations are not necessarily 'Letters to the Editor' about some article but small comments to a specific part of an article. They should either be public or private. Special kinds of annotations may have a priceless value to the reader and the editor of the library. To give some examples:

Rating of Content: Annotations can be used to rate some article. If this feature is easy to use, users will work with it and after some users rated the article, a preselection of articles for other users will automatically occur. Several discussion systems like news server and appropriate clients already implement that kind of preselection of articles. One can see this rating process as a 'review' process of the users using the library.

Link to the Future: Comments are not just text based. HTML tags are also allowed in annotations and therefore links to some external sources (identified by some URL) are possible. If the system has access to some citation index these 'Links to the Future' can be created automatically by simply looking for documents referring to that specific paper. The question 'Who uses this paper?' can be answered by search engines and if the result of that question can be interpreted by the system, links might be inserted automatically. On the other hand if one article is cited in a paper on the same server system, the cited article might be linked to the reference section of the paper automatically and the user have direct access to the cite.

Highlight / Cancel: Several programs (e.g. PC-Bibliothek [PCB, 2001]) are supporting this feature of highlighting some text in some text-document. This is also a kind of annotation (an annotation probably without content but with several attributes like colors, rating, time of annotation etc.). Again, if this feature is easy to use and several persons highlight or cancel some text, an automatic summary can be generated by the system and reading speed will increase enormous! This human based text mining will be a powerful feature. Problems will occur when implementing this feature. One will be the lack of programming support of the different web browsers. When using Java Script as a common basis (supported by most of the browsers) it is possible to extract the value of the selected text part, but it is not possible to get the position information! Some tricks and assumptions must be performed (e.g. assume, that the reader is highlighting the article in some 'known' sequence) to get the exact highlighted part of the HTML document.

Another problem will be the handling of different formats of one article. In J.UCS one paper is published in different data formats like PostScript, PDF and HTML. At the moment there are tools available for annotating HTML documents. The annotation feature is not available in PostScript-format documents, but in PDF documents. Therefore it must be possible, if one annotates the HTML-document at some point that this annotation is also available to the user in the PDF document using e.g. Acrobat Reader. Due to the fact that HTML is not a page description language it will be difficult to determine the exact position of the annotation in the PDF document especially if graphics are included in the document.

Questions and Answers: A special type of annotation is a question/answer dialog. As mentioned in [Maurer, 2001] this kind of annotation enables the implementation of 'active documents', documents which are able to automatically answer questions. In [Heinrich and Maurer, 2000] the basic idea of these documents is described: "The basic idea is that in the future, users of documents in any networked system should not just be able to communicate with other users, but also with documents."

[Maurer, 2001] explains some requirements and shows the power of many users. If hundreds of thousands of persons are working with a document, many semantically equal questions will arrive. To give a description of the work flow: The document is published at the system and facilities are available to ask any question at any position in the document. Experts answers such questions. These answers and the related questions are recorded.

> Experimental evidence shows that no new questions (just may be old questions in a new form) are asked after some 500 - 1000 viewers. Thus, if a document is viewed within, say, a year by 200.000 users, experts to answer questions are only required for a short time (until the first 1000 persons have visited the document). After this, all other 199.000 users get their answer from the system, if only the system can tell whether two similarly looking questions are indeed semantically identical. ... In our example, only 0.5% of users get their questions answered by experts, the remaining 99.5% receive the answer instantaneously from the system, a good approximation to the vision mentioned, indeed!

As one can see the old idea of annotations is a powerful feature especially when many users are working with the library. Obviously the tools must be available for every user on any client system and as discussed above these tools must be as easy to use as a pencil. Annotations are not the only possible interactive feature. One can imagine links from some very specific technical terms in the text to some dictionary server. Some systems like the one used at Learning Network of 'The New York Times' [Lea, 2001] have already implemented this useful feature.

Some large dictionaries like PC-Bibliothek [PCB, 2001], Brockhaus Multimedia ([BMM, 2000], one of the largest available multimedia collections) are client-side implementations. If one wants to use a dictionary, he simply selects the unknown string, type a defined hot-key combination and the dictionary does its work. Therefore changing the laptop implies also changing of the desktop-environment and the loss of the installed dictionary (like in the traditional world, if one changes the office, all books needed must be transfered too...). The client/server paradigm makes the live more easier for some users.

Implementing links in the content to these dictionaries shows, that user preferences are an essential requirement to realize this feature. An example will clarify this statement: Imagine two users of the library, a novice student of computer science and a professor of computer science. It might be nice, if there is a link from the term 'Programming Language' to some basic computer science dictionary for the student, but it makes no sense to create such a link for the professor! On the other hand if there are some new developed programming languages which are maybe interesting for the professor, then a link will also be suitable for the professor, but the target is then obviously a different document as for the student. This implies, that not just 'skill/experience' (e.g. in a range from 1 to 5) but also 'interest' is a criteria for the link-creation tool. Either the entry of a dictionary or the whole dictionary as such must be classified in a similar manner, i.e. a 'Basic Dictionary Of Computer Science Terms' might be declared to be of interest for students (skill/experience ≤ 2) and for very interested people (interest ≥ 4). Note, that it is absolutely necessary to evaluate the response of the users and to classify the dictionaries properly.

Currenty the Dublin Core Education Working Group (DCEd) proposed a Working Draft to the Dublin Core Usage Committee of the Dublin Core Metadata Initiative (DCMI) to overcome this problem [Mason and Sutton, 2000] of describing the audience for some material. They introduced a element called 'dc-ed:audience'. E.g. the Education Network Australia (EdNA, [EDN, 2001]) is in the process of implementing some vocabulary. Various education-specific thesauri contain audience classification descriptors (e.g., the Thesaurus of ERIC Descriptors and the NICEM Thesaurus, to name but a few) [Mason et al., 2000]. A discussion about different kinds of links (reference links, glossary links etc.) can be found in [Schmaranz, 1996].

One interesting topic currently discused at the World Wide Web Consortium (W3C) is the Semantic Web [W3C, 2001]:

> The Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.

Several techniques are available to describe data. If there is a common sense of the semantic of that meta data a Semantic Web can be implemented. This web can be used to find related articles or a cluster of similar articles automatically. In J.UCS similar articles are clustered in different collections. As described above, the ACM-classification scheme is used for the clustering of documents stored on

the same server system. If the Semantic Web becomes reality also documents stored on different systems can be clustered together and this would be a powerful feature for all users.

4 Conclusion and Future Work

There is a need for users to bias the layout and the content of a digital library. These user dependent preferences must be stored at some place. If these preferences are stored on the server-side there are enormous advantages. A workspace for a registered user of the Journal of Universal Computer Science (J.UCS) was introduced and some functions like rearranging articles, public and private bookmarks, and a global history were described. A digital library should not just be read, but a user should work with it. Links to the future, special annotations etc. are necessary tools to make full use of the power of many users. Reuse of existing material in terms of reference linking instead of cut-and-pasting is a key issue.

The next step is to redesign some existing annotation tools to make rating of content and other described annotation facilities possible and to implement some of the described functions. Articles are already described by some meta-data but these data are in a proprietary format and must be translated to a common format (like Dublin Core) to make the Semantic Web possible.

References

[BMM, 2000] (2000). Brockhaus multimedia. `http://www.brockhaus-multimedia.de`.

[MyB, 2001] (2001). `http://www.mybookmarks.com`.

[EDN, 2001] (2001). Education network australia. `http://www.edna.edu.au`.

[Hyp, 2001] (2001). Hyperwave information server. `http://www.hyperwave.com`.

[Lea, 2001] (2001). Learning network. `http://www.nytimes.com/learning`.

[PCB, 2001] (2001). Pc-bibliothek. `http://www.iicm.edu/pcbibliothek/home`.

[ACM, 1998] ACM (1998). Computing classification system. `http://www.acm.org/class`.

[Berners-Lee et al., 1998] Berners-Lee, T., Fielding, R., Irvine, U., and Masinter, L. (1998). Uniform Resource Identifiers (URI). RFC 2396. http://www.faqs.org/rfcs/rfc2396.html.

[Clarke, 2001] Clarke, R. (2001). Cookies. `http://www.anu.edu.au/people/Roger.Clarke/II/Cookies.html`.

[Heinrich and Maurer, 2000] Heinrich, E. and Maurer, H. (2000). Active documents: Concept, implementation and applications. *JUCS*, 6(12). `http://www.jucs.org/jucs_6_12/active_documents_concept_implementation`.

[Krottmaier and Maurer, 2001] Krottmaier, H. and Maurer, H. (2001). Transclusions in the 21st century. In *Proceedings of WebNet 2001, Orlando.* to appear.

[Mason and Sutton, 2000] Mason, J. and Sutton, S. (2000). Education working group: Draft proposal. `http://dublincore.org/documents/education-namespace/`.

[Mason et al., 2000] Mason. J., Sutton, S., and Co-chairs (2000). Education working group: Report of deliberations. `http://www.ischool.washington.edu/sasutton/dc-ed/Dc-ac/DC-Education_Report.html`.

[Maurer, 1996] Maurer, H. (1996). now hyperwave — the next generation web solution.

[Maurer, 2001] Maurer, H. (2001). Beyond classical digital libraries. In *Proc. NIT conference.*

[Maurer and Schmaranz, 1994] Maurer, H. and Schmaranz, K. (1994). J.ucs - the next generation in electronic journal publishing. *JUCS*, 0(0). `http://www.jucs.org/jucs_0_0/j_ucs_the_next`.

[Nelson, 1987] Nelson, T. (1987). *Literary Maschines.* Ted Nelson.

[Schmaranz, 1996] Schmaranz, K. (1996). Professional electronic publishing in hyper-g. In *Proceedings of WebNet 1996.* `http://curry.edschool.virginia.edu/aace/conf/webnet/html/130.htm`.

[W3C, 2001] W3C (2001). Semantic web activity. `http://www.w3.org/2001/sw`.