

# More than Passive Reading: Interactive Features in Digital Libraries

Harald Krottmaier

Institute for Information Processing and  
Computer Supported new Media (IICM)  
Graz University of Technology, Austria  
hkrott@iicm.edu

Denis Helic

Institute for Information Processing and  
Computer Supported new Media (IICM)  
Graz University of Technology, Austria  
dhelic@iicm.edu

**Abstract:** At the moment it is not very common to actively work with material stored on an Internet server system. Once the right information resource is found, only passive information consumption of documents is provided by a majority of information systems. While this is appropriate for a kind of reading material, this is a limitation for scientific articles. Scientific articles are often read more than once, are usually annotated while they are explored, and are often the root of further investigations. In this paper we explore two features which will be available in the Journal of Universal Computer Science (J.UCS). These features will support the reader in exploring articles and will make it much easier to turn information into knowledge.

## Introduction

Digital libraries are often the source of information. Three common steps are necessary to turn information into knowledge:

1. Find the right information.
2. Explore this information.
3. Apply the information.

Several tools are available to support scientists in finding the right information: search engines (such as Google) and directory services (such as Opendirectory) are often used as 'entry to the web'. While search engines are often used by users *who know exactly what they are searching for* and directory services are used by users *who do not know exactly what they are looking for*, there are also some system specific notification mechanisms available, which 'directs' users to sources of information. In the context of digital library systems this kind of service is called 'Alert Service'. These tools are currently well known and heavily used in the Internet community.

A few highly sophisticated features (such as 'find similar information resources') are already integrated in search engines and are inherently available in directory services. Other specialized application specific features are available in the server systems providing the information. In this paper we are going to discuss two features: an annotation feature and a personalization feature. We illustrate implementation details of these features. Annotations are available in J.UCS ([J.UCS, 2002]) since the beginning (1994). The personalization feature is currently under development and will be released in the near future. In J.UCS we use a Hyperwave Information Server (HIS, [Hyperwave, 2001]) to make the implementation of these features as easy as possible. In the following sections we explore some aspects of these features.

## Annotations

It is very common to write handwritten annotations on paper or mark some regions of text as important or unimportant for understanding, especially in personally owned material ([Marshall, 1997]). “Dog-eared” books are *highly customized books* and the turned-down corner may be of great value to the “creator” while trying to find an interesting part of the book again! No matter what type of content (text or graphic) on whatever position may be annotated in a traditional library environment. It is obvious that it should be possible to annotate also material stored in a digital library environment.

Unfortunately not all digital library systems support the reader in the creation of electronic annotations although electronic annotations are much more powerful than “analog” annotations in a traditional environment. The following list illustrates features of electronic annotations:

**Private Annotations:** if the digital library system provides the user with an identification facility, it is very easy to support private annotations. Private annotations must be visible only to the author of the annotation.

**Shared Annotations:** if annotations are stored in an intermediate- or server system, it is possible that users share annotations. Therefore many users may explore the content by using annotations very quickly. There are many applications of this kind, especially if users are working in groups and/or are exploring online-available literature.

**Typed Annotations:** types like ‘Question’, ‘Answer’, ‘Problem’, ‘Solution’, ‘Rating’ etc. are needful in selecting whether to read or not to read the annotation or even the content. Additional attributes like ‘User who wrote the annotation’ may be very helpful.

**Rated Content:** A document may be rated by special annotations. Users may decide to read just articles with top ratings and may not waste their time in reading (obviously) poor written articles. Actual content (i.e. recently published content) must be treated in a special manner.

Nevertheless, some technical problems have to be solved and tools must be created to support users in writing and managing annotations. Let us now discuss some approaches and problems in conjunction with the annotation feature. There are different possibilities of the storage location of the annotations. Annotations may be stored on 1.) client-side systems, 2.) intermediate systems and 3.) server-side systems ([Krottmaier, 2001]).

Annotations in J.UCS are stored on the server-side, therefore they are easily accessible to all users or to a group of users. Annotations in J.UCS are 1.) indexed and therefore search able, 2.) interlinked via special link-types with the content and 3.) physically separated from the content.

We mentioned the term *information source*. This kind of source is a collection of documents which all represents the same content. I.e. if content is available to users in different formats (e.g. in PDF, PostScript and HTML) we talk about *one* information source but different formats of the content.

This fact introduces another problem: Annotations to the HTML-representation of the content should also be presented to users when looking at the PDF-version of the content and vice versa. On the other hand, not every reader of the PDF-version is able to create annotations because the annotation feature is not available in Acrobat Reader. The Adobe Acrobat API-license prohibits development of plug-ins supporting annotations. Therefore another solution without using the PDF-inherent annotation facility must be developed. Currently we explore two approaches to solve this problem: feedback forms for *each page* in the document and hyperlinks from *each section* to an annotation server application.

There are many applications of annotations. One application is the ‘*active documents*’ concept ([Heinrich and Maurer, 2000]). Active documents introduce a system which may automatically answers questions. Users may ask any question at any position in a document stored on the server system. The system either answers the question automatically (by searching for answers to similar questions), or forwards the question to the author or maintainer of the document. If there are many users of the library and if the tools are easy to use, there is an enormous power in using this idea of active documents! But also other applications are possible: e.g. an *intelligent automatic summary* of a document. This summary can be as easy to implement as collecting and displaying all highlighted document fragments

which are created by the readers of the document. Again, many users may increase the quality of these summaries.

There are many studies about how annotations are written in an electronic environment (e.g. [Marshall et al., 2001]). Many web based environments (e.g. [Rosenstock and Gertz, 2001], [Kahan et al., 2001]) are using annotations. Annotations are a necessary feature in every system providing information to a large community.

## Rearrange Documents and Document Fragments

Organizing whole content entities or parts of content in a user defined way is essential when improving the usability of digital libraries ([Krottmaier, 2001]). We are currently implementing such a *server-side personal data collection* in the environment of the Journal of Universal Computer Science. This makes it easier for readers *to work with the content* of the library. In the upcoming prototype it will be possible to simply select information sources (in our case articles and article collection) stored on the server and link them to a personal collection. As a result of the integrated link database, the collections will remain up-to-date even if objects are moved to some other location on the server. Therefore the infamous 'HTTP-error 404' ('Page not found') cannot appear when accessing objects stored in the database.

Such a personalized data collection may be static or dynamic. The example above showed a static collection, where each article must be selected and linked to the collection. If there is the request to visualize 'all articles by author XXX' or 'all articles in category YYY', this approach is not very effective. Therefore other possibilities must exist: to answer the first question (articles by author), so called *query objects*, i.e. objects, representing a search query in the database, will be used. When these queries are executed, all matching objects will appear in the listing of the personal collection. The second question (articles by category) can be answered by linking the appropriate category-collection organized by the administrator to the personal collection of the user.

Since we are using an object-oriented database system (Hyperwave Information Server, HIS) which supports inherently users and groups, it is possible to implement *public* and *private* collections of articles simply by assigning the proper 'rights'-attribute. Public collections may be used as a kind of *public bookmarks* already implemented by a lot of services (e.g. [MyBookmarks, 2001]). But there are several other advantages, including accessibility, when saving bookmarks on the server-side rather than on the local file system in some browser-specific format.

Organizing objects stored *on* the server system (*local objects*) is obviously not sufficient in the context of an open digital library system. It must also be possible to organize objects stored on any other system (*remote object*) like HTTP-server systems, other internet based systems, and even files stored on the local file system.

To enable structuring of remote objects a surrogate object must be created in the database. This object (in Hyperwave terminology also known as 'remote object') is then used as handle for further operations. Since the control of the referenced part is completely up to the remote server administrator, tools (such as link checkers and content watchers etc.) must be integrated to support the user while working with remote objects.

Structuring a single document as one unit is easy compared to the task of structuring parts of a document. Resources are usually addressed by URLs (Uniform Resource Locators, [Berners-Lee et al., 1994]). Depending on the document format, more granularity is possible.

We should not limit this discussion of fragments of a text based document (such as HTML, XML and PDF). A user may address the upper left part of an image, or the first 5 seconds of a video, etc. There is much work to do, to implement this feature.

## Conclusion and Future Work

In this paper two necessary features were described in the context of the Journal of Universal Computer Science. Problems and issues of the well known annotation feature were presented and ideas on how to personalize documents and document collections were discussed.

In the current implementation it is not possible to add annotations *at any position* in the HTML representation of the information resource. Only whole content objects may be annotated at the time of writing. Although this approach is suitable for small documents, it is inappropriate when working with large documents. Therefore future work is addressed to make it possible to add annotations to parts of documents – simply by selecting a corresponding document fragment. Sharing annotations between different document formats is also an issue in the prototype.

The current working prototype does support rearrangement of locally stored objects, but does not support restructuring of document fragments. The next step is therefore to organize and restructure parts of documents.

## References

- [Berners-Lee et al., 1994] Berners-Lee, T., Masinter, L., and McCahill, M. (1994). Uniform Resource Locators. available online <http://www.w3.org/Addressing/rfc1738.txt> .
- [Heinrich and Maurer, 2000] Heinrich, E. and Maurer, H. (2000). Active Documents: Concept, Implementation and Applications. *Journal of Universal Computer Science*, 6(12):1197–1202. [http://www.jucs.org/jucs\\_6\\_12/active\\_documents\\_concept\\_implementation](http://www.jucs.org/jucs_6_12/active_documents_concept_implementation).
- [Hyperwave, 2001] Hyperwave (2001). Hyperwave Informatino Server. <http://www.hyperwave.com>.
- [J.UCS, 2002] J.UCS (2002). Journal of Universal Computer Science. <http://www.jucs.org> .
- [Kahan et al., 2001] Kahan, J., Koivunen, M.-R., Prud’Hommeaux, E., and Swick, R. R. (2001). Annotea: An open rdf infrastructure for shared web annotations. In *Proc. of the WWW10 International Conference*.
- [Krottmaier, 2001] Krottmaier, H. (2001). Improving the Usability of a Digital Library. In Hübler, A., Linde, P., and Smith, J. W., editors, *Electronic Publishing*, pages 178–182, Canterbury, Kent, United Kingdom. International Council for Computer Communication (ICCC) and International Federation for Information Processing (IFIP).
- [Marshall, 1997] Marshall, C. C. (1997). Annotation: From paper books to digital library. In *ACM DL*, pages 131–140.
- [Marshall et al., 2001] Marshall, C. C., Price, M. N., Golovchinsky, G., and Schilit, B. N. (2001). Designing e-books for legal research. In *Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries*, pages 41 – 48. ACM Press.
- [MyBookmarks, 2001] MyBookmarks (2001). <http://www.mybookmarks.com> .
- [Rosenstock and Gertz, 2001] Rosenstock, B. and Gertz, M. (2001). Web-based scholarship:annotating the digital library. In *Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries*, pages 104 – 105. ACM Press.