THE NEED FOR SHARING USER-PROFILES IN DIGITAL LIBRARIES

HARALD KROTTMAIER
Institute for Information Systems and
Computer Media (IICM)
Inffeldgasse 16c, A-8010 Graz, Austria
email: hkrott@iicm.edu

Personalization was a hype in the late 1990s. Several organizations offered (and are still offering) personalization services for their customers. From e-commerce applications we know that people are willing to provide their names and other personal data if they know exactly how this information is used. Many of us are using different Digital Libraries on a regular basis. If our interests change, we have to update all profiles stored at these different server systems. This task is time consuming and error prone. Since there is no single Digital Library which covers all information-resources a user needs, there is a need to exchange personal data (especially personal interests). In this article we will show that user-profiles are important to satisfy users' needs, which information about users should be stored in profiles. Just parts of the profile must be shared with other service providers. It will be shown, that it is not possible to predict usage of properties. Therefore the user itself must decide which properties may be shared with others and which properties should not be accessible by certain services.

Keywords: user profiles; creation of profiles; user-models; metadata about users

1. INTRODUCTION

Personalization is a very common concept in the development of stand-alone or web-based applications. It should tailor the interface *and* content to enable users to work in an optimal way with the system. Most service providers do not distinguish between "customization" and "personalization". However, in our discussion the difference of these two concepts must be distinguished and therefore let us explain it with the help of examples.

In the context of customization, the system *exactly* delivers information or services *requested by the user* either implicitly or explicitly. Personal interests are expressed by the user. The system is then able to deliver exactly the type of information and type of services the user wants to receive. This is a kind of "very simple personalization". To give an example: If a specific user wants the system to present navigational aids at the left side rather than the right side of the screen, this is a type of customization. If the user express interests in specific topics (aka "favorites") by selecting different attributes provided by the system, this is also called customization.

Personalizing is much more than customization. The system *assumes* what other information may be of interested to the user because of the interests that were expressed by the user or were expressed by another user. To give an example: If *user A* is interested in *topic X and Y*, and *user B* is interested in *topic Y and Z*, than the system may decide that *user A* is also interested in *topic Z*. One can imagine, that users of "type student" may be interested in similar topics. Therefore it is very common to suggest pieces of information to similar types of users.

Given the examples above it is obvious that customization is required to enable personalization.

It had been shown in many studies (see e.g. [1]) that users are willing to provide information about themselves, if they know how the information is used by the service-provider. If users do not know how this information is used, more that 80% of them are *not* willing to provide personal data.

In the field of Digital Libraries users actively want service-provider to exchange parts of the provided data! Since there is no single Digital Library which covers all information-resources users need, there is a need for exchange of personal data (especially personal interests). Many of us are using many different Digital Libraries on a regular basis. If our interests change, we have to update all profiles stored at these different server systems.

In the next section of this paper we will discuss requirements for personalization frameworks and techniques. Frameworks usually gather information about their users. Having collected necessary information either by profiling users or form-based question/answer dialogs, available content may be adapted (or "personalized") by the system for each user. Users should control at any time which data is stored at the server-side and how this information is used. Since parts of collected data are shared with other systems it is obvious, that users must be able to determine which parts should be shared. We will extend the list of requirements to make sharing of personal information as easy as possible.

2. ASPECTS OF PERSONALIZATION

This section provides an overview of aspects in personalization. As already mentioned in the previous section, personalization is about tailoring content and services to individual users. It is therefore obvious, that a system using personalization must:

• know it's *content and services*

• know it's *users* Personalization is either performed at single information-server-systems or at so called *portals*. A portal is an "one-stop entry" to the information-space providing a single interface to the user. This includes placement of navigational aids as well as placement of services and content.

Let us first describe why customization is necessary for users of Digital Libraries. Thereafter it is easier to understand why an extension to customization (i.e. personalization) is essential. We may benefit from cognitions from traditional libraries.

Two "features" of traditional libraries (TL) are well known: the *physical arrangement* of books and *annotations in printed material*. In the following discussion the location of a hyperlink pointing to the title page of an electronic entity corresponds to the location of a book on the bookshelf. The electronic content in DLs corresponds to information resources in TL like books and printed journals.

One very obvious customization feature in TLs is the arrangement of books on the shelf. A user inserts new books to the *right* position. "Right" is, wherever the user likes to put the book. This is completely up the user to determine this place. One may sort books by category, author, date of purchase *or* whatever is appropriate to the user. It is not possible in TLs to place one book (physically) at two places. Librarians developed representative cards which indicates that a book in this category is available but physically stored on a different place.

In DLs one entity (book, journal, article or any other content) may be accessible by more than one locations. DLs offer several indices (authors index, index of publication date, title index

etc.) to access the content of the electronic entity. It is not very common to let the user arrange electronic entries in a user defined way (although some systems like the ACM-DL – [2] – supports the user e.g. with "bookshelves".

Remember your first day in your local library? You where probably impressed of all the large bookshelves and all the index-cards. Searching a book by author was easier performed in the authors index than in front of the bookshelf. Nevertheless, finding a book *again*, if you are not aware of the author or the title of the book, is still easier done in front of the shelf, simply because you remember the location, where the book was stored at your last visit ("somewhere in the upper right..."). Additionally you remember the color and size of the book. These are very important properties of books (and any other information resource) which should not be neglected by software designers. The majority of electronic interfaces to Digital Libraries simply ignore these properties and hide them behind a blue (or any other color) link.

Much research had been done by many people so let us just mention the "libViewer"-project (see e.g. [3]). "libViewer" is a visualization tool for library entries. It visualize some properties like the size, color and age of the book, last usage of the book etc.

The second point we want to discuss is the interaction of the reader with printed material. It is very common to write annotations on paper or mark some portions of the text as important or unimportant for understanding, especially in personal owned material (e.g. [4]). Dog-eared books are highly customized books and the turned-down corner may be of great value to the "creator" while trying to find some interesting part of the book again! No matter what content (text or graphic) on whatever position can easily be annotated by the user in the TL. However, fast searching in annotations is impossible in a TL. Sharing annotation – a very powerful feature in the learning process – is very inappropriate in TLs. This kind of "destructive" customization is inappropriate in books borrowed from the public library. Nevertheless, annotations are a very important feature.

Exploration of these two features of libraries (location of information resources, and annotations) showed, that customization in a Digital Library is necessary. Let us now take a look at some implementation details and requirements for customization-services.

Customization is not for free. The system as well as the user have to perform some tasks to enable a simple customized presentation of content and services. The involved parties – articles and other content objects as well as users – must be *profiled* in some way. There are different kinds of content objects in a Digital Library: "traditional" library content, like books, articles, table of contents etc. and modern Digital Library content, like annotations, structure facilities etc. These objects must be clearly distinguished to enable information filtering.

An article consists of the content and some attributes (i.e. metadata) of the content. In ordinary server systems an article is represented by some file where the content is stored. The format of the content depends on the used publishing system. If the format of the file is HTML, arbitrary metadata like authors name(s), title of the document, date of publication etc.can be stored in HTML META-tags in the file. In that case content and metadata of the content are stored in the same file. In stricter document formats like Portable Document Format (PDF, [5]) only a very restricted and predefined set of metadata can be stored inside the document (document author, subject of document, creation date etc.). Older document formats like PostScript [6] do not inherently support metadata. Therefore it is necessary to store all metadata in a separate entity.

Different standards and recommendations about the format of the metadata are available. Initiatives like [7] and [8] are pointing out the importance of metadata. Modern information management systems like Hyperwave [9] separate metadata and content to alleviate management

of data.

As noted above, profiling the content is absolutely necessary to enable customization. The system have to know certain attributes of the content. This kind of profiling is done by adding specific metadata to the content. Customization in a Digital Library environment takes not just a fixed set of metadata and the content itself into account but relations between objects in the library.

The question "What topics are of interest to user of the system? " must be answered to enable customization. Currently two approaches (as well as combination of those) are used:

Implicitly: Every single click of the user is tracked by the system and the targets are recorded. The system is then able to spot out the part in that the user is interested. This approach is useful in large collections of material and in very long customer relationships. [10] reported about this technique as tool for personalization. Web usage mining and click-stream analysis supply much information about user behavior. Nevertheless, starting a personalized service is difficult because no information about user's behavior is available to the system. Therefore the next approach is often used to start a service:

Explicitly: Questionnaires must be filled out by the user. The answers are transfered to the system. The system then have enough information to present selected topics of interest.

Depending on the offered content-types attributes differ. To give an example: A library of films will have attributes like: "User A is interested in films directed by director X and performed by some actor Y". In a digital journal, keywords in titles and/or abstracts are more relevant.

The content presented to the user must match the profile or must be manipulated to match personal preferences. These preferences are not just valuable to users, but also to publishers. The publisher can improve the offered services if popular topics are known. When storing the profile of the user – independent on how the profile was created – privacy must be considered to protect users.

Different customization mechanisms used in different systems makes it very difficult to reuse user-profiles. Some solutions are available for specific Digital Library services – like Hermes, a notification service for Digital Libraries [11] – and a generic pool of user profiles for different service provider will probably exist in the future.

Having the content profiled and the interests of the user stored, the server system is able to adapt content and/or interface for the user. In the next section we take a look at what a user wants to customize in a Digital Library environment.

3. ABOUT USER-MODELS

User-models are abstract representations of users. One can see implementations of user-models as "metadata about users". User-models must be effective and efficient, but most important: the model must be accepted by the user. Acceptance is achieved by supporting users in many respects.

In [12] the aims of user-models are explored in detail. Let us here summarize and comment the results.

active information providing: the user-model should describe topics of interests of users. Therefore the system may actively inform the user via some notification service about new and related information. This technology is known as *push-technology*.

reduce information overload: a study [13] produced by faculty and students at the School of Information Management and Systems at the University of California at Berkeley shows the current information overload. The team attempted to measure how much information is produced in the world each year. While in the year 2000 the amount of unique information was between 1 and 2 exabytes, the amount was about 5 exabytes in 2003. 92% of the new information was stored on magnetic media, mostly in hard disks. Film represents 7% of the total, paper 0.01%, and optical media 0.002%. This is about 800 MB per person. It is not possible to implement a personalization concept (e.g. described in [14]) without the use of user-models. It is obvious that users are not interested in reading or browsing through unrelated material. They want to retrieve information related to *their* interests. Implementing user-models may help in reducing this enormous potential information overload.

support of handling: depending on experience and/or education of the user, the system may help with appropriate hints.

improvement of queries using users-knowledge: if the number of result-objects is too high (or low), the system may change the query-expression to create an optimal set of results. To give an example: when too may objects are found by the search-engine, they may be filtered by e.g. preferred authors of the user etc. to reduce the number of objects. If the query is to narrow it should be widen by the system automatically.

improvement of queries using expert-knowledge: the system may use profiles of experts in a specific topic to improve search-queries, e.g. by adding related terms — which are unknown to the user — to the query.

ranking of results: search-results are usually ranked because of relevance to the query-string. Results may be clustered related to interests of the user. If user A is interested in topic X, Y and Z then the results should be grouped in these 3 topics.

team-support: to extend the knowledge of a singe-team member it may be of benefit to other members to know about the "knowledge-space" (i.e. user-profile) of other members in the same team.

In the next paragraphs we summarize properties to be stored in a user-profile (and therefore must be modeled in a user-model). It is clear, that many properties must be available in the context of groups. Reflecting on the categories of data, it becomes clear that users are not interested in sharing provided information with any other party.

personal data: nickname, firstname, lastname, date of birth, gender, education, and

other related information. Similar data for groups. **interests:** obviously users (or group) interests are of interest to the system. This includes keywords of topics, clusters of related classification-systems (e.g. ACM-classification system, [15]) etc.

personal preferences: this includes e.g. color-schema to use and customization of

different services provided by the system.

personal experience: if the user is an expert or novice with the system itself. How

much help and which functionality should be provided by the system. The category "personal data" transparently shows problems about access for other service providers to attributes described there. Attribute "nickname" or "gender" may be accessed, but users may not allow certain information- or service providers access attributes such as "date of birth" or "education". Therefore a very flexible architecture must be available for users to express their needs.

[16]explored a similar model for users: Information needs of users differ with respect to their type, their content, and their duration. Since these needs are different between users, all aspects have to be taken into account during user modeling process. Amato and Straccia described a five categories model (personal data, gathering data, delivering data, actions data and security data category). Let us look at one specific category, the *gathering data category*. It collects preferences and restrictions about the documents a user is looking for. They found three sub-categories:

• document content category, i.e. what type of content has to be gathered by the system, e.g. text written in a specific language.

• document structure category, properties related to the structure of all documents to be retrieved by the system, e.g. electronic document formats, type of documents (articles, news, poems), creation date etc.

• document source category, i.e. sources of documents such as authors, publishers etc. An interesting aspect in the previously cited work is the proposed usage of the information gathered by the system implemented in the *actions data category*.

A personalized service should be highly responsive to the needs of the user. In particular, long term information needs involve repeated interactions with the user. Assuming that a lot of the user actions are consistent, a retrieval service should match increasingly better his/her needs over time. Furthermore, since the interaction could extend over a long period of time, it cannot be assumed that the users interests will remain constant. The change in interest could be anything from a slight shift in relative priorities to completely losing interest in some domain and gaining interest in another. In general, a system must be able to detect or must allow the user to indicate the change in interests and should respond by adapting to these changes. The system must be able to explore newer domains and prospect for interesting information. To summarize, personalized service should be capable not only of dealing with the currently known needs of the user, but exploring different domains to find documents of potential interest to the user. Thus, it should be specialized, adaptive and exploratory.

Questions arise when thinking of actual user-profiles: Who is the owner of the profile? Is it the service provider, who collected the data, or is it the user itself? In our point of view the user is the owner of the records (i.e. the profile). The user should decide, how this data is used by service (or information) providers. Unfortunately current systems rarely respect this circumstance.

Expressing privacy practices of service providers [17] in an automatic processable way is already available and implemented in browsers.

The Platform for Privacy Preferences Project (P3P) enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents will allow users to be informed of site practices (in both machine- and human-readable formats) and to automate decision-making based on these practices when appropriate. Thus users need not read the privacy policies at every site they visit.

Although P3P provides a technical mechanism for ensuring that users can be informed about privacy policies before they release personal information, it does not provide a technical mechanism for making sure sites act according to their policies. Products implementing this specification MAY provide some assistance in that regard, but that is up to specific implementations and outside the scope of this specification. However, P3P is complementary to laws and self-regulatory programs that can provide enforcement mechanisms. In addition, P3P does not include mechanisms for transferring data or for securing personal data in transit or storage. P3P may be

built into tools designed to facilitate data transfer. These tools should include appropriate security safeguards. As mentioned above, in our opinion the user is the owner of the profile and therefore the user must express explicite willingness, which sets of properties should be accessible by other users or service providers.

4. IMPLEMENTATION OF USER-PROFILES

Using standard metadata models for user-modeling such as Dublin Core (DC, [7]) is not be a solutions to the discussed problems. As mentioned in the previous sections, it should be possible to *restrict access to certain properties* of a user-profile. DC was not intended to provide such a restriction. DC consists of 15 elements such as title, creator, subject, description etc. where values are assigned. DC introduced a "Rights"-Element in DC-3. This element should describe rights to the document described by the DC-metadata-set but not to single properties or attributes of the metadata-set. The situation is similar to other implementations of metadata schemas.

RDF (Resource Description Framework, [18]) in contrast is much more flexible. An expression consists of the following triples: *subject*, *predicate* and *object*. These triples are encoded using XML. Subjects are resources to be described. Since predicates as well as objects may be subjects and may therefore be described via predicates and objects, this approach is very powerful and new concepts may be created by any user. In the literature the triple (subject, predicate, object) is often expressed as (resource, properties, statement). Adding simple "allow" or "deny" statements will make it possible to restrict access to certain properties.

5. CONCLUSIONS

It had been shown that it is necessary in the field of Digital Libraries to share user-profiles. Currently metadata of content is shared with other providers (see e.g. the Open Archive Initiative) but there is little discussion about sharing of user-profiles. Users do not want to share *all* attributes or properties of their profile therefore it should be possible to add right-attributes to single entries in the profile.

ACKNOWLEDGMENT

This work is supported by DELOS, a Network of Excellence on Digital Libraries (EU FP6, G038-507618).

REFERENCES

[15] ACM (1998). ACM Computing Classification System.

[2] ACM Digital Library (2004). http: //portal.acm.org/dl.cfm (2004/03/31)

[6] Adobe (1989). *PostScript Programmiertechniken*. Addison-Wesley Longman, Inc.

[5] Adobe (1993). *Portable Document Format Reference Manual*. Addison-Wesley Longman, Inc.

[16] Amato and Straccia (1999). User Profile Modeling and Applications to Digital Libraries Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries (ECDL 1999), Springer

[7] Dublin Core Metadata Initiative (2004) http: //dublincore.org (2004/04/07)

[11] Faensen, Faulstich, Schweppe, Hinze, and Steidinger (2001). Hermes - A Notification Service for Digital Libraries. In Proceedings of First ACM/IEEE-CS Joint Conference on Digital Libraries, Roanoke Virginia, USA.

[9] Hyperwave (2004). Hyperwave Information Server. http: //www.hyperwave.com (2004/03/21)

[14] Harald Krottmaier (2003). Stop Reading (Useless Parts)! Proceedings of the 7th ICCC/IFIP International Conference on Electronic Publishing (ELPUB 2003)

[13] Lyman and Varian (2003). How Much Information http: //www.sims.berkeley.edu/how-much-info-2003 (2004/04/07)

[12] G. Möller (1999). Brevis: Benutzermodelle in IR: interne Repräsentation, Erstellung und Visualisierung Master Thesis, Oldenburg

[4] Marshall (1997). Annotation: From paper books to digital library. In *ACM DL*, pages 131–140.

[8] Open Archive Initiative (2004). Open Archive Initiative. http: //www.openarchives.org (2004/03/30)

[17] World Wide Web Consortium (2004) Platform for Privacy Preferences Project (P3P) W3C, 2004 http: //www.w3.org/P3P/ (2004/03/29)

[1] New Survey Shows Consumers Are More Likely to Purchase At Web Sites That Offer Personalization (2001) Personalization Consortium http: //www.personalization.org/pr050901.html (2003/12/03)

[10] Pierrakos, Paliouras, Papatheodorou and Spyropoulos (2003). Web Usage Mining as a Tool for Personalization: A Survey in: User Modeling and User-Adapted Interaction, Kluwer Academic Publishers

[18] Resource Description Framework (2004) RDF Interest Group and Semantic Web Activity Group http: //www.w3c.org/RDF (2004/04/12)

[3] Rauber and Bina (2000). Visualizing electronic document repositories: Drawing books and papers in a digital library. In *Proceedings of the 5. IFIP 2.6 Working Conference on Visual Database Systems (VDB5)*, pages 95 – 114.