A Practical Architecture for User Modeling in a Hypermedia-Based Information System[1]

Julita Vassileva[2]

Universität der Bundeswehr München
Institut für Technische Informatik
85577 Neubiberg, Germany
E-mail: jiv@informatik.unibw-muenchen.de

2

Abstract*

User Modeling is a field of increasing importance for industrial applications, especially for information retrieval from large data-bases using browsing as a search strategy. The most of the research in this field, however, has been theoretical.

We have implemented a new architecture for user modeling based on analysis of the tasks performed by the users. It allows adaptive browsing support for users with different levels of experience, data-protection, and a degree of adaptability according to the preferences of individual users. This architecture was applied in building a user modeling component for a hypermedia-based information system for hospital information which is now being experimented.

Introduction

Browsing is a useful technique for retrieving documents from data-bases (Thompson & Croft, 1989). It has been widely applied recently as hypertext and hypermedia systems have become increasingly popular (Begoray, 1990). The main cognitive advantage of this technique is that the users in general are better able to recognise the information they want than to characterise it in advance. The disadvantages of browsing are that it is easy to get lost in a complex network of nodes representing documents and concepts and that there is no guarantee that a browsing search will be as effective as a more conventional search. If it offers a rich set of links, the system is responsible for helping the users understand what the links mean, how they might be used, and where they are in the network formed by them. Without this kind of help, browsing can take on the aspect of the user finding his way in a maze, where he can become hopelessly "lost in (hyper)space" (Conklin, 1987). User Modeling can help in supporting the user's navigation (Kobsa, 1993).

Our work stems from an industrial project for creating a large hypermedia information system for hospitals. After a long phase of interviews and observations we came up with an architecture of a User Model (UM) which :

- supports novice users by ensuring a smaller browsing space, while providing to experienced users a larger browsing space and possibilities for direct access to the required information;
- combines user modeling with data-protection;
- provides adaptiveness and self-improvement;
- provides adaptation at discrete points in time, after the user has been informed;
- provides tools for the users to adapt their user models;
- supports collaborative work.

Application Domain

HYNECOS is an information system for hospital information developed by Siemens AG in co-operation with the University Clinic of Orthopaedics in Heidelberg (Hertwig, 1993). It contains multimedia data about patients, personnel, hospital stations (room-plans, beds, and occupancy) and medical concepts. All this information is organised logically following the Hypertext Development Methodology (HDM) for creating hypertext from relational databases (Grazotto et.al, 1991). The system is implemented in ToolBook on an IBM PC 486. HYNECOS is still at a prototype stage, it contains only the necessary minimum of information to demonstrate the abilities of the system. Even during initial testing of the prototype, it became clear that user modeling would be of a major importance for the success of the system, since the group of potential users of HYNECOS was very broad and heterogeneous and the amount of information to be offered (browsing space) was too big.

After some initial discussions with users we conceived the idea that the UM should be set in the context of the user's tasks, so that these could be used as a basis for "filtering"

* This work has been partially supported by Siemens AG Corporate for Research and Development and the Bulgarian Ministry of Science and Higher Education under contract I-24.

Published in Proceedings of the 4th International Conference on User Modeling, Cape Cod, MA, 14-19 August 1994, 115-120 **(*best paper award *)**

information. We created a general scheme of a task-hierarchy. With this empty scheme, we interviewed one user who filled it with the tasks which he typically performs and the information he needs for them. With this example-scheme in hand, we interviewed several other users whom we considered as typical. They easily interpreted the empty scheme according to their specific tasks. In this way we obtained several schemes reflecting different tasks and information needs. We used these schemes to refine our idea of the architecture of the system and the UM. The process of interviewing helped us draw some conclusions about the restrictions and requirements for user modeling in hypertext-browsing for practical applications.

Specific Problems of User Modeling in Browsing Information Retrieval

Acquiring data about the user

Most of the methods for acquiring data about the user's interests, reported in theoretical research papers cannot be used in our application. In principle, the task of finding out the plan or goal of the user in browsing is more difficult than in query-based information retrieval (Kok, 1991): the user's browsing activities can be chaotic and non-sequential and they do not necessarily reflect his goal or task. That is why all known adaptive interfaces for browsing use as evidence other aspects of the user's behaviour that might reflect his interests. For example, (Kaplan, et al., 1993) assume that the more time is spent on a unit, the more interesting it is. In other approaches (Thompson & Croft, 1989), (Kok, 1991) the user is asked to estimate the "interestingness" of every unit that is retrieved. In our case, as will be explained later, neither of these methods is realistic.

A lot of other restrictions are posed in a practical application. For example, almost all theoretical approaches treat the hypertext system as modifiable on the basis of the information in the UM, if this is required (Kaplan et al., 1993). However, in our case the hypermedia system is a "given" and it can be only "masked" or "viewed".

User group identification

Different approaches for representing a UM for browsing exist: some take a symbolic, logic-based perspective (Kok, 1989), others use connectionist schemes, like associative networks (Belew, 1986), (Kaplan et al., 1993). However, in our case, where there are clearly identifiable categories of users, it is worth taking advantage of this fact. There is no point in using techniques that are appropriate mainly when nothing is known about the interests of the user and only a strongly individual user model can be helpful as in (Belew, 1986) and (Kaplan et al., 1993).

The presence of user groups reduces the need to find evidence about the interests of individual users. This suggests the use of some kind of stereotype model (Rich, 1989), (Chin, 1989). Stereotype approaches, however, are not so widely used for modeling the user's preferences in information retrieval. The main difficulty in our case is that it is hard to give one systematic classification of users, because the factors influencing the information needs are many — for example, the task performed, the place, the profession and the rank of the user — and combining these factors requires a weighting scheme, i.e. giving higher priorities to some of the factors. The only general solution will be to represent explicitly all factors and their possible values in order to ensure a coverage of all possible combinations.

Task-based context for user modeling

In our domain, as in many other application-domains, different user groups have typical tasks and goals for their data-retrieval. For example, (Kaplan et al., 1993) also assume a fixed set of user goals; they don't infer the goals from the user's browsing activities. Every goal (task) has specific information needs that provide a context in which the information (topic) needed is known in advance (Tyler & Treu, 1989). In our case, however, the tasks appeared to be decomposable, i.e. hierarchically organized, and their information needs are mutually dependent. That is an important difference from the approach of Hyperflex (Kaplan et. al., 1993), where the goals are represented as independent nodes in the hypertext structure, at the same level as the nodes, corresponding to the hypertext-topics (see figure 1). This means that every new goal, introduced in the system is considered to be semantically independent of all other goals, i.e., the weights of the links from this goal to all hypertext-topics have to be given explicitly (or learned by the system). In our approach, a new task will be considered first with respect to the other tasks in the hierarchy so that its place can be found, and certain information needs can be ascribed to the task (at least the information needs of its children nodes). This means that comparatively less knowledge engineering effort is needed for assigning weights to the goal-topic links (we therefore do not provide the machine learning capability of Hyperflex).

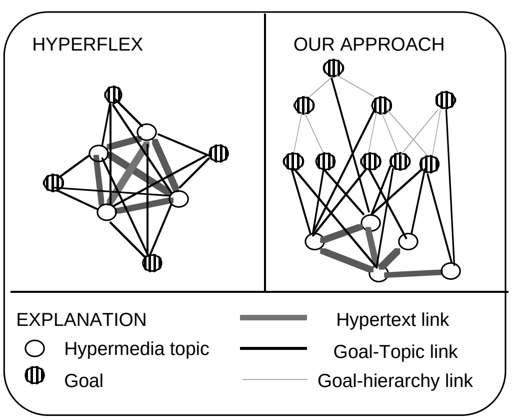

Figure 1: The goal (task) representation: two approaches.

Direct access vs. browsing

In some previous studies — e.g., (Thomson & Croft, 1989), (Kaplan et. al., 1993) — it has been assumed that the user doesn't know exactly what he is searching for, because it is new information, e.g., news or advice. In our domain, by contrast, the user normally knows what document he needs. The question for him is only how to obtain it conveniently. In principle, for such an application, a query-based information retrieval ought to be ideal. But the experience of the clinic with the same documentation represented in a relational database showed that the users experienced serious difficulties when formulating queries; the hypermedia-based prototype had a far higher degree of acceptance. For the real application, however, the chains of documents to be visited became too long and the choices offered were sometimes too confusing. Unlike (Dumais, 1988) and (Furnas, 1986), we decided not to merge the browsing paradigm with a query-based but rather to rely solely on user modeling to provide sufficiently direct access to information.

The user's level of experience with the system

There was a disagreement among users during the interview phase about whether it is better to have a smaller number of directions in which to search at a given time, even if this is restrictive, or always to be able to access the desired information directly. Users who had not previously seen the system preferred to have a small set of links available at any moment, so that making each choice would be easier. Users who knew the system better, felt comfortable with the unrestricted interface provided by the prototype which had no UM.

These two requirements conflict, and a compromise can be found only if we explicitly represent the factor "experience" as a way to determine the degree of direct vs. browsing access to the information. We had to choose whether to make the system *adaptive* with respect to this factor — i.e. to create means for the system to infer the level of experience of the user from his browsing — or to make it adaptable — i.e. to provide means for the user to change the level of experience in his UM when he wants. We decided to implement both decisions.

Too high adaptation is not always an advantage

The users need to have a coherent mental model of the system. A system that is constantly adapting, even if this is supposed to be happening for their own benefit, makes them feel uncomfortable and decreases their confidence. We came to this conclusion when we observed users working on different versions of HYNECOS. We believe, the policies usually applied in connection with new software releases (which are announced in advance) is accepted by users much better: They need to know what is going to be changed in the system and why. In the medical domain it is sometimes of vital importance to access data quickly; the users therefore want to be able to have absolute confidence in their system. Another reason not to strive for maximal individualisation is that the system also serves communicative functions. For example, it can be used on the same computer by two or three doctors and several nurses. In order not to confuse and impair the communication between them, the system should look the same for users performing the same tasks. That is why the design of the UM should provide for creation and adaptation by a group of users (not necessarily a homogeneous user-class) which is going to work together in a team. In other words, the system should support collaborative work by means of a "group user model".

Data-protection

With the already existing noncomputerised medical documentation, doctors, nurses and students were free to examine the files with the patient data. However, all of the interviewed users agreed that there must be data protection in the electronic version of the patient data, especially because far more people will have access to it. Data protection is an important issue which so far has not been considered in connection with user modeling in information retrieval. Normally, every information-rich database has different user rights of access. User modeling can help to ensure data-protection from unauthorised access.

An Architecture for User Modeling.

The proposed architecture for user modeling can be described as a three-layer structure to be added on top of the hypermedia system (figure 2). The first layer contains representations of the tasks performed by the users. They define the "views" of hypermedia and provide the main context in which a specific UM is situated. The second layer contains information about the user classes. It provides specific constraints on the rights of access to information and requirements for the form of presentation. The third layer contains the individual user models. Every individual UM contains additional information not implied by the user class, for example the user's level of experience.

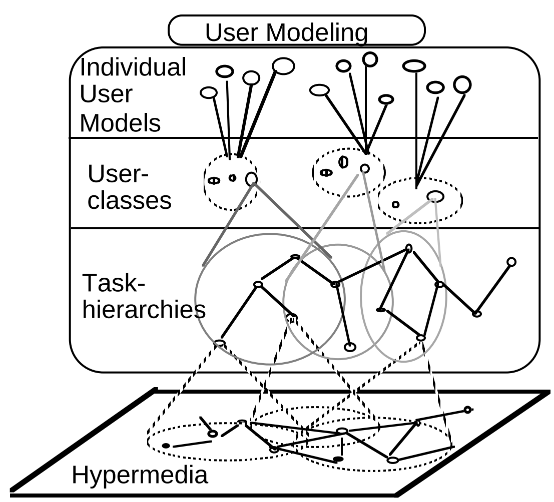

Figure 2: Architecture for User Modeling

Task-hierarchies

The typical tasks performed by the users that involve work with the information system are represented as hierarchies. Every task implies specific information needs, i.e. *entities* in the HDM - terminology (topics, nodes). When the user is performing a relatively specific task (lower in the hierarchy), he sees a limited view over the hypermedia, one which is relevant to the task; moving up in the task-hierarchy, he gets wider views (cf. figure 3).

Ways of defining views over the hypermedia.

A task-dependent view can be defined in two ways:

• "Free browsing with an anchor" — i.e., providing links to the entities needed for the task, and allowing the user to browse further following the standard hypermedia links from these entities. In this way, the task serves as a sort of anchor. It provides the starting points in browsing to which users can always come back, if they get lost.

• "Restricted browsing" — i.e., allowing the user to browse only within the entities linked to one task. In this way the normal hypermedia links outside of the view are disabled ("masked"). If the task is a high-level one, the browsing space includes the browsing space of its sub-tasks.

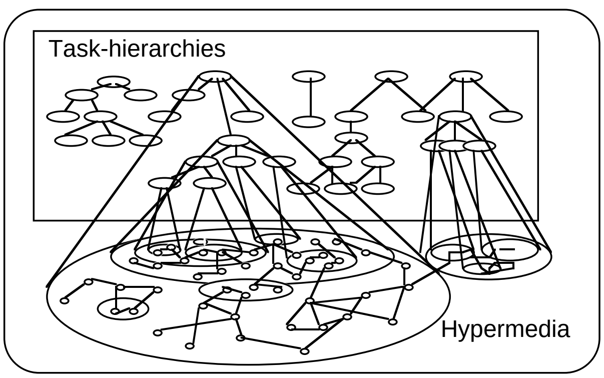

Figure 3: Architecture of the User Model.

We expected that this second way of viewing would be preferred by novice users as they would prefer to work on one task at a time. The experimental results with a group of novice users showed, however, that about half of them preferred to be able to browse freely, following the logical links of the hypermedia, provided they have an anchor to return to. We offer two possible explanations, which were informally supported in additional interviews and tests:

1) The larger part of the users were not actually executing the task they had chosen, but rather working simultaneously on several tasks. After switching to a task one-level higher, they no longer felt restricted in their browsing. This shows that the level of experience has not been set appropriately for them.

2) A small proportion of the users could be seen to be unable to formulate the information needs of the current task exactly. They were searching for information that they didn't actually need in order to complete the task.

It was recognised, however, by people from both groups that the restrictive way of defining a view improves task performance, saves unnecessary browsing and generally helps them organise their work better.

The question of which way of defining a view is better, can be related to the general question of the degree of the system's adaptation to the user with respect to the user's adaptation to the system. We left this question to be answered by the users, by allowing them to chose the viewing style themselves.

Task-determined rights to modify information

The tasks define not only rights to access, but also to modify information. One way to reduce the risk of data corruption and loss is to associate rights to modify data only with the tasks that are expected to change this data. For example, during the task of "Administering therapy", the nurse needs to know the patient's diagnosis, but the diagnosis is only supposed to be changed during performance of the task "Making a diagnosis". Other forms of data protection are implemented at the user-class level, as discussed in the next section.

User-classes

The users population can be divided into several overlapping user classes with different information needs, rights of access to information and appropriate forms of presentation. The factors that define the user class in our case are: profession (doctor, nurse, manager, student, or patient), location (ambulance or station), and rank (up to 5 stages depending on the profession). A user-class is characterised with a combination of values of these factors.

Every user class can be related to a different set of task hierarchies or isolated low-level tasks. Normally, there is an inheritance in the task hierarchies of classes with the same profession. For example, a chief doctor at the station has to perform all the tasks of a doctor plus the task of station management.

Rights to access and modify information are largely dependent on user class. Higher-ranking users may have two types of special rights:

1) The right to access a larger amount of data which is not included in the task hierarchy of the user class. Every user can access all data, except where access is explicitly prohibited for his or her user class.

2) The right to modify data even when this is not allowed by the task which is currently being performed.

Rights to access or modify data can also be restricted for certain classes of users. For example, patients are restricted to getting only information from their own files and from the medical concept base. Students are not allowed to see personal data and to modify any information.

The user class can imply also specific presentation needs. If there are several alternative representations of the same entity, one is chosen. that is considered to fit best the needs of the user class. Such presentation preferences can, however, be changed in the individual UM.

Individual User Models

A user class serves as a "kernel" to which many individual user models are related (see figure 2). Every individual UM contains parameters which specify the user's level of experience and his or her requirements with respect to the task hierarchy, the style of "viewing", the form of presentation, and the screen layout. Where some parameters are missing, the corresponding values are inherited from the user-class. The individual UM can be changed both by the user and by the system (after consulting the user) when nonoptimal performance is observed.

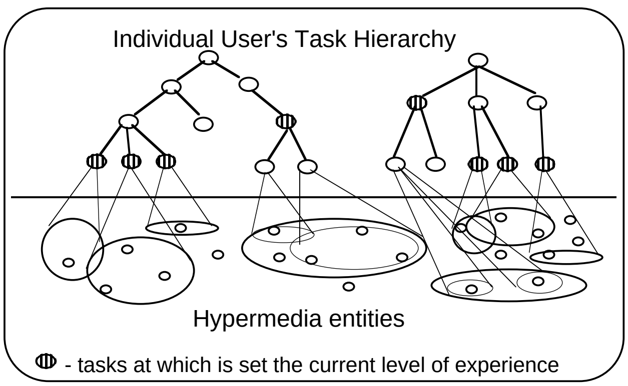

Figure 4: The user's level of experience.

Level of experience is a parameter of the individual UM which determines at what level of every branch of his task hierarchy the user can get access to the hypermedia-entities (see figure 4).

The specific rights of an individual user to access and modify data — which may differ from those determined by the user class — are determined by a set of parameters. The individual UM contains also parameters corresponding to the user's special presentation preferences and screen layout.

Basic Features of the System

Context

In HYNECOS, the context of interaction between the user and the system is defined when the browsing space is limited according to the task selected from the task-hierarchy. An experienced user can immediately get the view over the entire hypermedia system. A novice will be guided through the task hierarchy until he reaches a level that he has shown an ability to cope with; then he will be given the corresponding smaller view. By gradually increasing the navigation space together with his moving up in the task hierarchy, the user is always interacting with the system in an appropriate context.

Adaptation

Our interpretation of the notion of adaptation is slightly different from that of (Croft, 1984). We believe that the system must not only "make available those tools which are relevant to the current task", but also be able to change dynamically according to the changing user's needs (not only with respect to the tasks) in order continually to maintain the appropriate context for interaction. Strongly adaptive systems, however, threaten the user with a loss of control, and their users have difficulties in developing coherent models of them (Fischer, 1992). That is why we decided that the system's adaptation to the user's needs has to be carried out not continuously, but at discrete points in time, and only after the user has given specific permission.

Because one of the most important time-dependent factors is the user's experience (Norcio & Stanley, 1989), the system must have means for finding out and reacting to the changes in the user's level of experience. The user's navigation actions are recorded, and if patterns are found in them which imply that the user's proficiency has increased, a flag is set indicating that it seems appropriate to increase the user's recorded level of experience. For example, if he goes down the task hierarchy, reaches his current level of experience and, without browsing in the information space provided, immediately clicks on the "task completed" button, this is a sign that he wants to get to the higher level and be allowed a broader view of the hypermedia. When there is evidence that the user's recorded level of experience is too low (or too high), the system changes it, after obtaining the user's consent.

Data is also collected about particular types of representation that are retrieved especially often. After a threshold has been exceeded, the user is asked whether he really prefers the type of representation in question. On the basis of his answer, the parameters of his individual preferences are updated. Similarly, the system collects statistics about nonoptimal behaviour of the uses in different classes. This information is used for revision of the task-hierarchies and the links from the tasks to the information entities. In this way the system can improve its task hierarchies across time.

Adaptability and Group User Models

User-adaptable systems support users in modifying systems according to their own needs (Fischer, 1992). Adaptability is a typical feature of systems for adaptive browsing (Thompson & Croft, 1989), (Tyler & Treu, 1989), (Belew, 1989), (Kok, 1991), (Kaplan, et al., 1993). Our architecture for user modeling allows the user to adapt his individual model in the following ways:

• Modification of the individual task-hierarchy. The user can define a task hierarchy of his own. The user is supported here with a library of *task-aggregates* (i.e., parts of task-hierarchies and their information needs) from which he can cut and paste to alter his own task hierarchy.

• Creation of new tasks. The user can define his own tasks and add them to the library of task aggregates. For this purpose he is provided with means to select information entities from the hypermedia by browsing and to link them to the task which he wants to create. During the modification of the task-hierarchy and creation of new tasks, the rights of access to information of the user cannot be changed, since the "forbidden" entities are specified in his user class and are therefore invisible for the user. However, he can extend his browsing space and make it more convenient for search.

• Selecting the style of viewing. The user can select the "free browsing with an anchor" style or the task-restricted browsing.

• Changing the recorded level of experience. This can be done by the user explicitly, i.e. without waiting for the adaptation mechanism to suggest a change in the level.

• Changing the presentation preferences. The user can change the values of these parameters directly in his individual model.

An important consequence of the availability of tools for adaptation by the user is that it becomes possible for the users to build group models to support cooperative work. They have to find an agreement about the group task hierarchy, the style of viewing, the level of experience and the type of presentation.

Conclusions

User modeling is a field of increasing importance for industrial applications, especially for information retrieval form large data-bases using browsing as a search strategy. We propose an architecture for user modeling based on empirical analysis of the tasks performed by the users. It ensures adaptive browsing support for users with different levels of experience, data-protection, and a level of adaptability according to the preferences of individual users.

This architecture was applied in the development of a user modeling facility for a hypermedia-based information system for hospital information. Currently we are experimenting with the system with four different classes of users and the results are very encouraging.

Acknowledgements

I am grateful to Ralph Deters, Karin Hertwig, Dr. Krämer for helpful discussions, to the anonymous reviewers for their comments and to Anthony Jameson for helping with the English.

References

Begoray, J. 1990. An Introduction to Hypermedia Issues, Systems, and Application Areas. *Int. J .Man-Machine Studies*, 33: 121-147.

Belew, R. 1986. Adaptive Information Retrieval: Machine Learning in Associative Networks. Ph.D. Thesis, Dept. of Computer & Communication Sciences, University of Michigan, Ann Arbour.

Chin D. 1989. KNOME: Modeling What the User Knows in UC. In A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems, 74-107,Berlin-Heidelberg:Springer Verlag.

Conklin, J. 1987. Hypertext: An Introduction and Survey, *IEEE Computer*, 20: 17-41.

Croft, W. 1984. The Role of Context and Adaptation in User Interfaces. *Int.J. Man-Machine Studies*, 21: 283-292.

Dumais,S., Furnas,G., Landauer, T., Deerwester, S.,Harshman R. 1988 Using Latent Semantic Analysis to Improve Access to Textual Information, In Proc. of CHI'88, 281-285, New York: ACM.

Fischer, G. 1992. Shared Knowledge in Cooperative Problem Solving Systems: Integrating Adaptive and Adaptable Systems, In Proc. of UM'92, Dagstuhl, 148-161.

Furnas, G. 1986. Generalized Fisheye Views, In Proc. of CHI'86, 16-23, New York: ACM.

Grazotto, F., Paolini, P., Schwabe, D. 1991. HDM - a Model for the Design of Hypertext Applications, In Proc. *Hypertext'91*, 313-328, ACM Press.

Hertwig, K. 1993. Section D5.2. Medical Appliction, Deliverable D5 (Applications Specification and Design) Contribution to ESPRIT Project HIFI, Siemens, Munich.

Kaplan, C., Fenwick, J., Chen, J. 1993. Adaptive Hypertext Navigation Based on User Goals and Context, *User Modeling and User-Adapted Interaction*, 3.

Kobsa, A. 1993. User Modeling: Recent Work, Prospects and Hazards, Technical Report Nr. 26/93, Dept. of Computer Science, Universität Konstanz.

Kok, A. 1991. A Formal Approach to User Modeling in Data-Retrieval, *Int. J. Man-Machine Studies*, 35: 675-693.

Norcio, A., Stanley. J. 1989. Adaptive HCI: A Literature Survey and Perspectives. *IEEE Trans. on Systems, Man and Cybernetics* 19 (2): 399-408.

Rich E. 1989. Stereotypes and User Modeling: in A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems, 35-51, Berlin-Heidelberg: Springer Verlag.

Thompson R.,.Croft W. 1989. Support for Browsing in an Intelligent Text Retrieval System, *Int.J.Man-Machine Studies* 30: 639-668.

Tyler, S., Treu, S. 1989. An Interface Architecture to Provide Adaptive Task-Specific Context for the User, *Int.J.Man-Machine Studies* 30:303-327.