A User Interface and Knowledge Delivery Solution for a Modern WBT System

Denis Helic
Institute for Information Processing and Computer
Supported New Media
University of Technology Graz

Hermann Maurer
Institute for Information Processing and Computer
Supported New Media
University of Technology Graz

Nick Scerbakov
Institute for Information Processing and Computer
Supported New Media
University of Technology Graz

Abstract: In order to take a part in a particular Web-based training session learners need to work with a number of tools reflecting a particular training strategy. By operating these tools learners access different training objects contained in that training session. Such training objects contain relevant information for their current training task. Nowadays, modern WBT systems support dozens of different, sometimes rather complex tools and provide access to thousands of training objects. In this paper we analyze the problems of user interface that can become a rather complex one in training sessions conducted in such systems. In order to overcome such problems we provide a simple general user interface solution for these WBT systems. Further we provide an evaluation of the responses from users we gathered in applying that solution in a number of Web-based training sessions. This evaluation showed us that there is much more potential in our solution than we believed at the first resulting in the evolution of our user interface solution to a simple knowledge delivery tool. Such tool might be used to conduct training sessions that contain always the most up-to-date and the most relevant training objects residing in the system.

1. Introduction

Technically, WBT systems (Gaines et al., 1997) consist of a large repository of training objects and a number of tools to manipulate these training objects. Usually, these tools support operations such as creating, deleting, accessing, or updating of training objects (Dietinger et al., 1997).

According to the variety of types of training objects supported by a particular WBT system, or in other words according to the number of tools and functionality that such tools support we may distinguish between (Helic et al., 2000):

- Standard WBT systems
- Advanced WBT systems.

Standard WBT systems support only basic training objects, such as learning unit, learning course or discussion forum. Tools provided by such systems include authoring tools, publishing tools, navigation tools, simple search tools, etc. Usually, such tools are rather primitive with a simple basic functionality and a comprehensive user interface. On the other hand, advanced WBT systems support a wide range of different training objects. For instance, these training objects include learning goals, mentoring sessions, brainstorming sessions, knowledge cards, knowledge profiles, etc. Structurally these objects may be rather complex. For instance, a knowledge card is an instance of so-called semantic network, which is a rather complex structure. Obviously, tools provided by advanced WBT systems are by far, more numerous,

provide a richer functionality and more complex than tools provided by standard WBT systems. Consequently, these tools have a rather complex user interface.

From the users' point of view WBT systems offer possibilities to take part in different training sessions. Usually, a training session is considered to consist of a so-called training strategy and a number of training objects (Helic et al., 2001a). Training strategy reflects a particular way of working through the subset of training objects to achieve a particular training goal (Helic et al., 2001b). Actually, a training strategy is a collection of tools combined in a certain way and reflecting a particular training methodology. For instance, consider the training strategy known as Web-based learning. This training strategy is a collection of tools that provide functionality needed to access and navigate through different learning units and learning courses (Andrews et al., 1995). Web-based learning might be considered as the basic training strategy supported by all WBT systems. Another example of a training strategy would be so called Web-based tutoring. This strategy is a rather advanced training strategy supported by an advanced WBT system called WBT-Master. Web-based tutoring prescribes working with a set of special WBT-Master tools. These tools allow users to navigate through a sequence of so-called learning actions and training objects associated to such learning actions in order to achieve a particular learning goal.

Obviously, advanced training strategies supported by means of advanced WBT systems might consist of a rather large number of tools (Helic et al., 2001a; Helic et al., 2001b). Considering that such tools have rather complex user interface themselves any collection of these tools would have a rather complex user interface as well. Thus, users of advanced WBT systems might be confronted with a serious problem of a complex, inscrutable user interface. Also taking into account that more advanced training strategies might include training objects of many different types, thus resulting in an increased number of different tools needed to implement such a training strategy, the user interface problem is becoming even larger. Evidently, if we want to be able to conduct training session by means of advanced training strategies we need to solve the increasing user interface problem on a larger scale. Hence, not only do we need to simplify the user interface of a particular tool but we need also a more general user interface solution that is able to reflect all peculiarities of any training strategy and provide a single access point to each of tools needed to implement a particular training strategy. Apparently, such solution needs to be highly configurable, customizable user interface solution.

WBT-Master provides such general user interface solution in the form of so called Personal Desktop.

2. Personal Desktop

Conceptually, a personal desktop is just a set of folders containing references to designated training objects and WBT-Master tools. Tools from a particular folder reflect a particular training strategy, whereas the training objects from that folder contain information relevant for achieving a certain training goal.

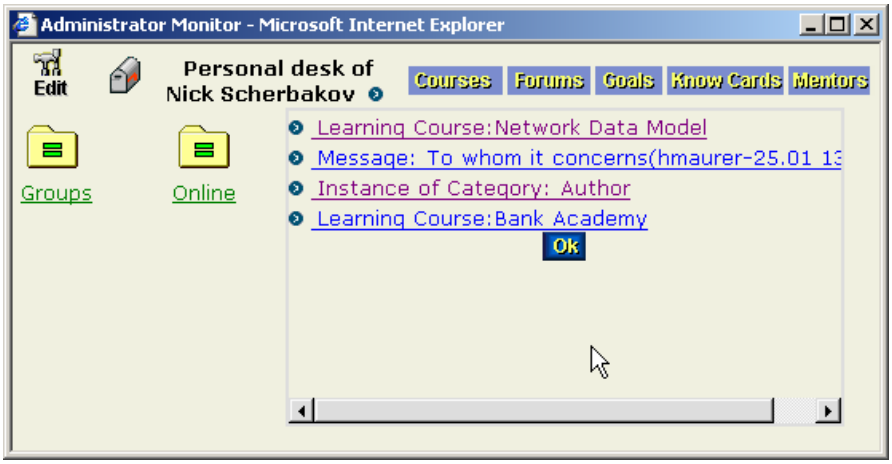

Figure 1: Content of a personal desktop folder

The main idea behind the concept of a personal desktop might be seen as the following. An advanced user, say an author or a tutor manages a particular training session with a group of learners.

He/she creates a personal desktop folder containing different WBT-Master tools and a number of training objects residing on the server. Once when the personal desktop folder has been created he may share that folder with the learners' group, thus allowing them to participate in his/her training session. Obviously, by accessing the created personal desktop folder learners have access to designated training objects by means of the implemented strategy. However, learners are not any more confronted with arbitrary tools or training objects provided by the system. Rather they are supposed to access only few relevant training objects and that by means of tools reflecting the desired training strategy.

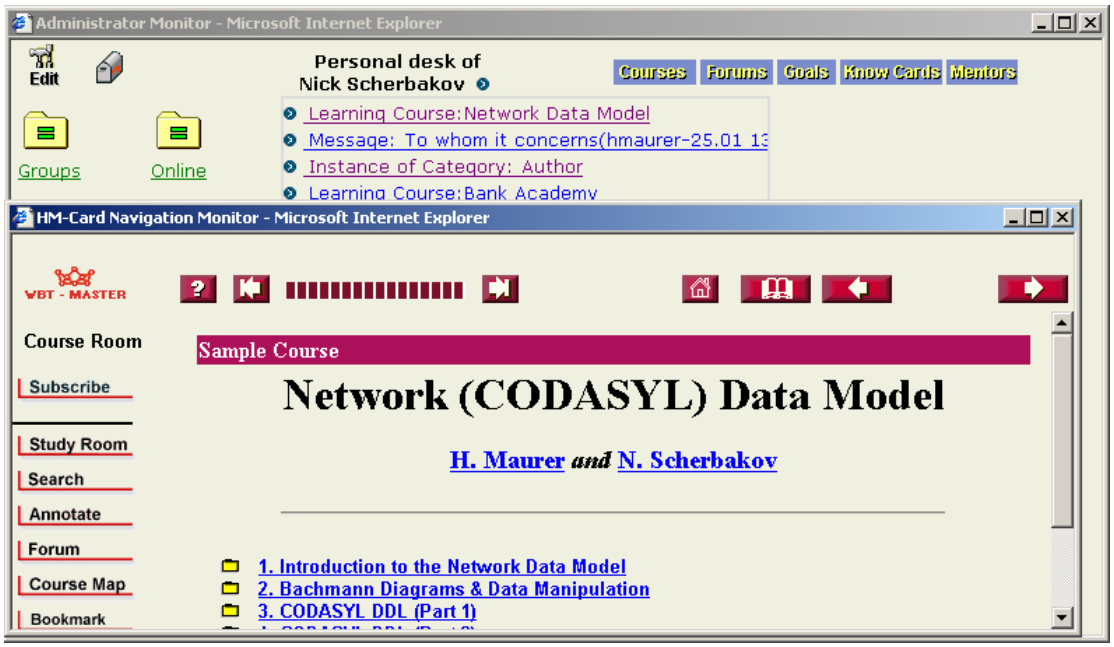

Figure 2: working with training objects from a personal desktop folder

This concept greatly facilitates the fact that a typical user works with just a few tools and training objects offered by the system. In other words, this special customization mechanism is used to adjust the rich system functionality to personal needs of a particular learner or a group of learners. Hence, a tutor or an author may decide on preferable user interface for such group of learners and on training objects, which are needed to accomplish a particular training task. Another important aspect of the personal desktop concept is collaboration facilities, which are provided by so-called shared folders and internal messaging system. For example, a learner group may share a certain folder to put all contributions of the group members into it. In this way, the contributions may be easily accessed by group members and discussed by attaching messages to such contributions.

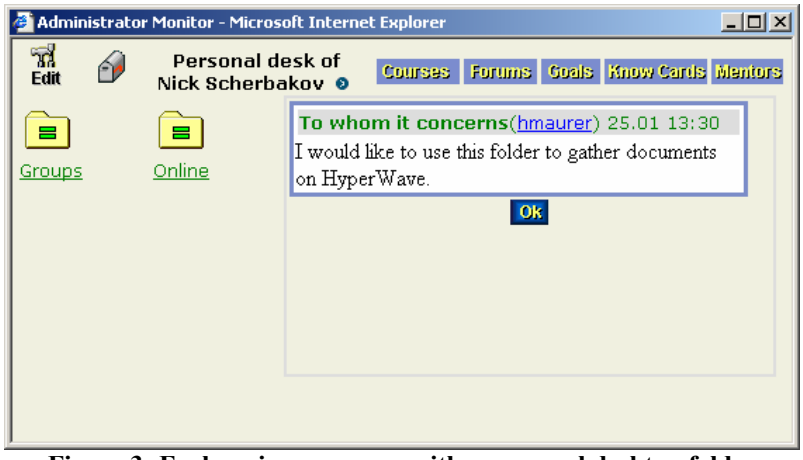

Figure 3: Exchanging messages with a personal desktop folder

3. Working with Personal Desktop

Generally, first experiments with personal desktop show a great acceptance of this concept by users.

Authors very much appreciated a possibility to define all requisites of a particular training session, i.e., all tools that are needed to implement a particular training strategy, as well as all relevant training objects as members of a folder. In this way they were not supposed to know all peculiarities of the system in order to combine the system tools into a coherent training strategy. For instance, they were not supposed to know how to attach a discussion forum to a learning course to provide learners with a board to discuss topics from that learning course, but rather they just put that learning course and a discussion forum into a personal desktop folder and the above-mentioned relationship between these objects was automatically established by the system.

On the other hand, tutors liked the messaging functionality of a personal desktop folder the most. This functionality allowed them to keep the communication between members of a particular learner group within the scope of a particular training session. Also, possibility to reuse (for instance, in the form of FAQ) the results of such communication was quite well accepted by tutors.

Finally, learners appreciated the simplicity of user interface as offered by such training sessions very much. Not any more did they need to find their way through dozens of system tools and large number of training objects. Rather they just worked with a few tools and relevant training objects combined into a simple navigable list.

However, there exist some disadvantages of this concept as well. The most important shortcoming of such an approach can be stated as follows. A particular training strategy, i.e., a particular collection of tools that implements that training strategy might be seen as a rather static entity. That means that a particular training strategy is likely not to change in the course of the time. For instance, consider the above-mentioned Web-based learning training strategy. This strategy consists always from say a number of learning courses and a discussion forum attached to these courses. What really changes in training sessions is training objects. Actually, training objects might be seen as a very dynamic entity. Not only that in different training sessions training objects are completely different but also in one and the same training session training objects are likely to change. For instance, some training objects might become obsolete, newly updated versions of training objects might be created, etc.

These results led us to the conclusion that we need to extend the concept of personal desktop and redesign it in a way that it becomes robust to such changes. Thus, we decided to incorporate a mechanism that would be able to automatically select training objects that are the most relevant and the most up-to-date training objects. Only such training objects should become members of a particular personal desktop folder. Obviously, such mechanism must be able to "reason" about training objects and decide whether a particular training object is relevant to a particular training session, i.e., does it match certain criteria posed by that training session.

4. Personal Desktop as knowledge delivery tool

WBT-Master supports semantic data structures, which can be used to create different semantic overviews of training objects residing in the system (Helic et al., 2001a). These structures include so-called knowledge cards and knowledge domains.

A knowledge card is a description of particular concept (i.e. semantic entity). For example, a semantic entity "Database technology" may be seen as a knowledge card. Practically speaking, each knowledge card may provide access to a number of associated training objects. For example, a course on "Relational Data Model" may be associated with the knowledge card "Relational Data Model", some other training objects may be associated with the same knowledge card. Knowledge cards may be interrelated into a semantic network using different types of relationships: "is a part of", "is a kind of", "synonym for", etc. For example, the knowledge card "Relational Data Model" may be related as "is a part of" to the knowledge card "Database Technology". The knowledge card "World Wide Web" may be related as "is a kind of" to the knowledge card "Hypermedia Systems". The knowledge card "Web Base Training" may be related as "is a synonym for" to the knowledge card "Computer Supported Collaborative Learning".

The infer mechanism essentially utilizes the other important property of the semantic network - a possibility to infer training objects using semantic relationships. Whenever users access a knowledge card,

the system automatically infers all training objects, which are associated with this particular knowledge card and with knowledge cards related to this one. This mechanism greatly facilitates the initial access to the most relevant training objects.

On the other hand, a knowledge domain is a collection of training objects, which are structured using a predefined template called the knowledge domain schema. A knowledge domain schema may be seen as a definition of semantic categories and all possible semantic relationships between them. Thus, a knowledge domain contains training objects, which might be seen as instances of a particular semantic category interrelated by means of semantic relationships with training objects that are instances of related semantic categories. Basically, we may use this mechanism to express facts such as: the document "C" (an instance of say "Module" category) is related to the document "A" (an instance of say "Author" category) by means of the "Author Modules" semantic relationship. Similarly, the document "C" is related to the document "B" (an instance of say "Project") by the means of the "Project Modules" semantic relationship, etc.

Now, users may browse and search knowledge domains by means of the terms defined by the knowledge domain schema. For instance, users accessing an instance of the "Author" category automatically get a link to an index of all instances of related categories, i.e., a link o all "Author Modules" appears on the screen. Similarly, knowledge domain might be used to execute semantic queries, i.e., to search for all "Modules" that are related to a certain "Project" training object, etc.

The extension of the personal desktop concept treats knowledge cards and knowledge domains as training objects that might be added to a personal desktop folder.

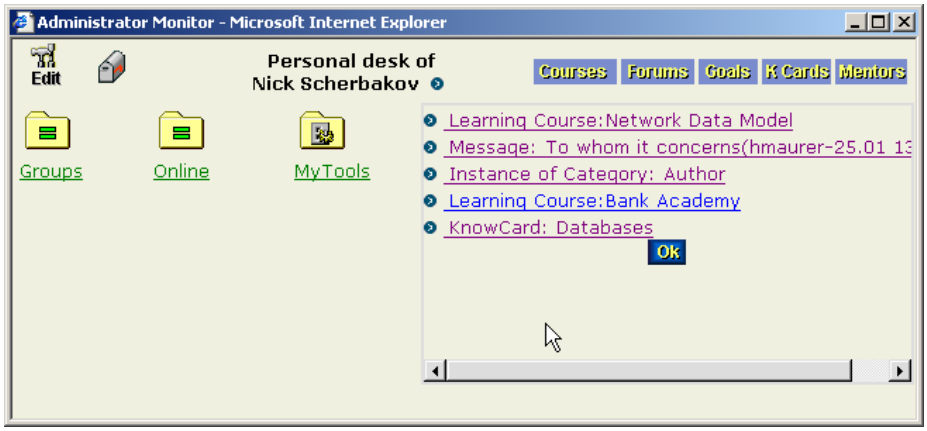

Figure 4: Adding a knowledge card to a personal desktop folder

By adding a knowledge card as a training object in a personal desktop folder we achieve the following. Whenever users access a knowledge card from a personal desktop folder the system automatically infers all related and to that particular concept relevant training objects.

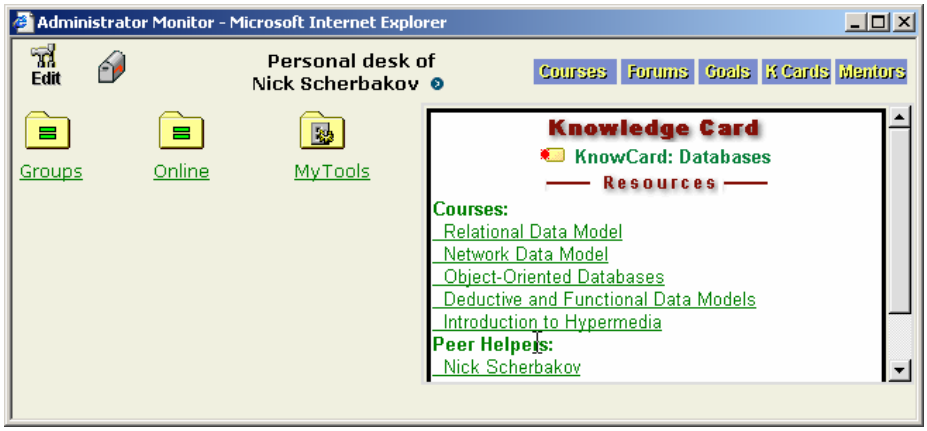

Figure 5: Accessing a knowledge card from a personal desktop folder

If a new training object is associated with this knowledge card, the system automatically captures that change by inferring the newly added training object the next time we access the personal desktop folder. Thus, this mechanism is a rather robust one to changes in the underlying repository of training objects. On the other hand, by adding a knowledge domain to a personal desktop folder users are provided with possibility of browsing semantic overviews of training objects residing in the system. Again, if a new training object is added to such an overview, the system automatically updates the overview the next time when we access it. Thus, changes in the system's repository of training objects are captured again. Hence, by incorporating these two simple knowledge-processing mechanisms into the concept of personal desktop we were able to enhance personal desktop from a general user interface solution to a simple knowledge delivery tool.

5. Conclusion

Generally, first experiments with the Personal Desktop system demonstrate a rather good functionality and acceptance by users. Learners like the situation where they are not confronted with too much system tools or irrelevant training objects but rather with just few important tools and training objects. On the other hand, tutors and authors very much appreciate the fact that they were able to completely control their training sessions and especially corresponding training strategy. They can be sure that their learners are confronted with only those tools that are the essential part of the implemented training strategy. Moreover, the extension of the concept of a simple user interface tool to a knowledge delivery tool leads to the conclusion that learners are not only confronted with the relevant system tools but also with the most relevant and up-to-date training objects that are needed for a successful training sessions.

References

Andrews K., Nedoumov A., and Scherbakov N.: *Embedding Courseware Into Internet: Problems and Solutions*, Proceedings of ED-MEDIA'95, pp.69-74., 1995

Dietinger T., Maurer H.: *GENTLE - (GEneral Networked Training and Learning Environment);* Proceedings of ED-MEDIA'98, AACE, Charlottesville, VA

Gaines, B.R. and Shaw, M.: *Institutional Transformations to a Learning We*b, in: Proceedings of ED-Media/ED-Telecom, the World Conference on Educational Multimedia/Hypermedia, T. Müldner and T.C. Reeves (Eds.), 1997.

Helic D., Maurer H., Scerbakov N.*: Web Based Training: What Do We Expect from the System.* Proceedings of ICCE 2000, Taiwan 2000, 1689-1694.

Helic D., Maurer H., Scerbakov N.: *Knowledge Domains: A Global Structuring Mechanism for Learning Resources in WBT Systems*, Proceedings WebNet 2001, AACE, Charlottesville, USA 2001, 509-514.

Helic D., Maurer H., Scerbakov N.: *Mentoring Sessions: Increasing the Influence of Tutors on the Learning Process in WBT Systems*, Proceedings WebNet 2001, AACE, Charlottesville, USA 2001, 515-519.

Teege G. and Koch M.. *Integrating Access and Collaboration for Multimedia Applications*. Proceedings of International Conference on Multimedia, Hypermedia and Virtual Reality, Moscow, Russia, pages 170--176, P. Brusilowsky, editor, September 1994.