In Montgomerie, C., Viteli, J.(Eds.) Proceedings of ED-MEDIA 2001. Norfolk, USA: AACE. pp. 1023-1028

Scenario-based Design of Flexible Hypermedia Learning Environments

Huberta Kritzenberger, Ronald Hartwig, Michael Herczeg

Institute for Multimedia and Interactive Systems
University of Luebeck
Seelandstrasse 1a
D-23569 Luebeck
{kritzenberger, hartwig, herczeg}@imis.mu-luebeck.de

Abstract. Software development processes need to cover usability relevant information. Although several methods have been proposed how to collect usability relevant data, there is a lack of suggestions on how to deal with this information and how to ensure that it is available in a proper way at the relevant stages of the software development process. This paper proposes a framework to store and retrieve usability relevant data using object-oriented OOA and OOD methods together with scenario-based analysis. As this framework combines advantages of all these methods, uses a XML-based, relational database with server pages and networking, it is an ideal repository for distributed design teams.

1 Introduction

The focus of this paper is on usability engineering in the development process of adaptable web-based course material. Such applications are under development at our institute for the domains of e-business and web-based education training material for virtual universities (cf. Hartwig/Kritzenberger/Herczeg 2000).

As hypermedia are used by users with many different goals and levels of knowledge (cf. Kritzenberger/Herczeg 2000), there is a need for systems that allow to modify and adapt parameters accordingly. Although adaptability has been a long time the primary domain of artificial intelligence research (cf. Brusilowsky 1998) there is a growing interest of usability engineers (De la Teja/Longré/Paquette 2000). They see part of the success of developing software adaptable to the individual user's needs and preferences depending more on managing the development process and integrating the assessment data produced at each stage of this process than on increases of information technologies.

Therefore, much attention has been paid to the software development process with several attempts to improve it and to integrate usability information into it, e.g. ISO 13407. Ideally, the usability engineering (UE) starts with the definition of the potential users and an analysis of their work situation in order to specify design requirements. The analysis phase focuses on understanding users and processes needed for their work and the result of the analysis should be documented by means of user-centered work-modeling techniques. Several techniques are available to do that, like survey technique similar to market research, contextual research (cf. Holtzblatt/Beyer 1993) or participatory design. Regardless which technique is actually used for analysis, normally a lot of data are produced, as demonstrated for example for contextual research. The data produced in one phase can hardly be recorded in appropriate ways and integrated or reused in other software development phases. Therefore, it does not make a great difference for the practice of software development processes, if these data are gathered or not. Data have to be usable by different members of a design team at a different stage and with different backgrounds. Furthermore, the members who are active at different stages of the development process will raise new data, which add to the conceptual design model and should therefore be available, too.

2 Knowledge Management for Usability Engineering

Usability engineering (UE) is an iterative process of development, which consists of six stages: analysis, requirements specification, design, implementation, evaluation and maintenance. In all these stages data are produced which are also needed in the following stages by different members of the software development team. In our approach all members of the design team, which in the development process of learning environments consist of content author, designer, producer and quality manager, use a relational database with a web front-end. This database should include all the context information as well as the content of the teaching units. In a first step this information is only presented to the involved persons but no contents are automatically generated.

For database organization object oriented techniques are used (cf. Hartwig/Kritzenberger/Herczeg 2000). Classes of information like “user attributes”, “organizational requirements”, “content”, etc. are identified and then may be freely combined into “views”. These views are role dependent, for example the content author may concentrate on the analysis data and the contents. Whereas the producer, who has to implement the learning unit, may need additional design rationales added by the designer. A quality manager can base his evaluation of the system on the requirements identified and documented during the analysis phase. As the size and the number of the learning units is difficult to handle, techniques of the OOA are used, e.g. abstraction, inheritance and generalization. If for instance many different user groups are to be considered, they may be ordered hierarchically: “All users”, “users of the teaching unit”, “A special group within these user group”. Attributes of the most general object (“All users”) are inherited by all following objects and then, e.g. the designer’s view includes all attributes, from the general to the specific ones. OOA techniques, like underspecification and refinement make the handling of large object sets easier and allow all participants to start with rather raw data and to refine them during the iterative process. Using OOA techniques based on a database has several advantages. The database supports the complete lifecycle of the course unit and makes all information and design rationales available again for maintaining or updating the course. As the contents of the database are XML(XHTML)-based they can be included into the courses with less manual effort. Furthermore, the connection of each unit to the position and role in the database is kept. Additionally for each production phase there is appropriate additional information available, like design rationales and the related context information. Updating a content in the database automatically updates the course and avoids inconsistencies.

3 A Task Analysis and Design Framework

The following framework layer represents a generic model (cf. Herczeg 1999, Herczeg 2001), which covers the specific characteristics of the user population as well as the situational factors and the process of using the software, e.g. a hypermedia learning environment. It is based on the concept of object-oriented system design. It was developed in the context of interactive applications (cf. Herczeg 1999) and proves more and more to be a generic platform for analysis and design of software systems.

The following figure shows the task analysis framework, which is a set of object classes from which a model of the system can be built. It enables building a functional as well as a contextual model. It uses the basic ideas of object-oriented analysis and design (OOA, OOD), but enlarges the object classes with extra attributes covering context of use information, e.g. describing the conditions of use, the environment, goals etc.

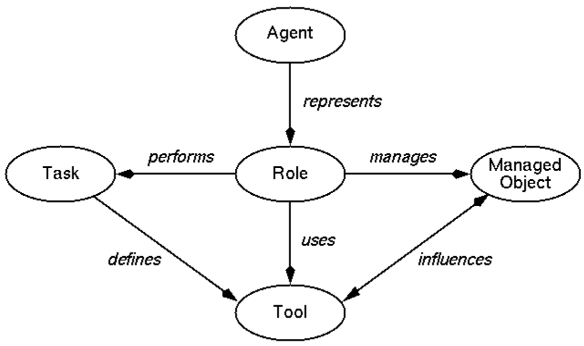

Figure 1: Task analysis framework (from Herczeg: 1999:26)

The framework consists of the following entities:

Managed Objects*:* defines the resources managed by the system, e.g. for the domain of educational material these are the modules of the application and information space.

In Montgomerie, C., Viteli, J.(Eds.) Proceedings of ED-MEDIA 2001. Norfolk, USA: AACE. pp. 1023-1028

Tasks: have to be performed by those agents, who use the system according to their role. For the domain of the learning environment tasks are typically performed by learners as learning tasks, e.g. doing exercises, trying out, memorizing something, constructing and so on). Tasks define tools, which help or enable executing the task. In educational material (like web-based training systems) the task can also be systematically specified by a learning theory.

In our example of educational material this means that the task can be described as “learning” at the highest level of abstraction. In the sense of a refinement hierarchy, at a more detailed level, subtasks may be described as “basic orientation”, “acquire knowledge”, “apply knowledge”, “transfer knowledge”, and so on. At the most detailed level the model may describe atomic tasks. This leads to the idea of a refinement hierarchy of the task in which the behavior of the system is described at increasing levels of detail. The lowest level of detail is that of atomic pieces of behavior. In the direction of the aggregation hierarchy the system has subsystems at the next level of aggregation, which are themselves part of a compound system at the next higher aggregation level.

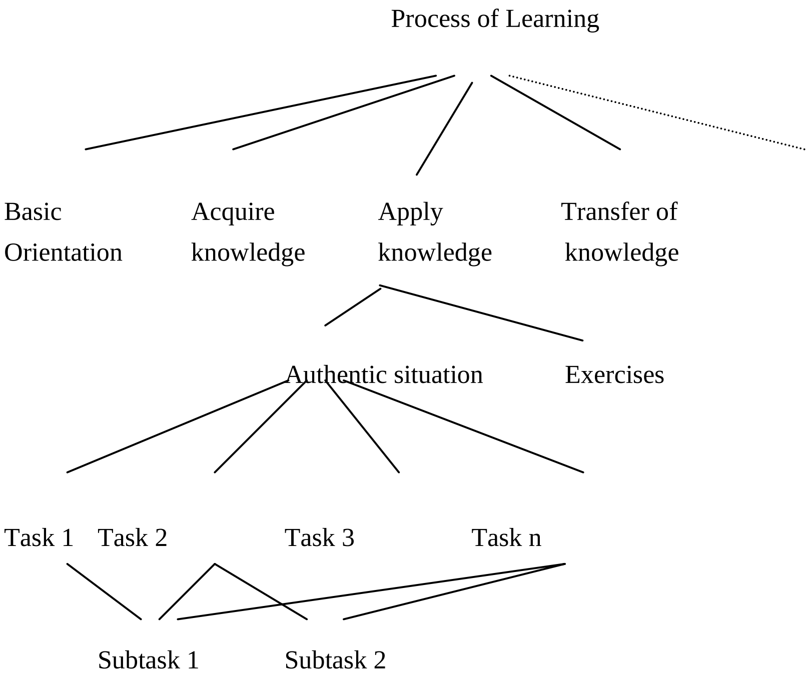

Figure 2: Example of Task Structure for the learning process

Roles*:* are represented by agents. In the case of educational systems this means that the role is represented by agents, who are the learners.

Agents: describes users and their profiles, which may be active in one or more roles. Learners may be diverse, but learners may also have several characteristics in common and can therefore be divided into several user groups. In this case role would be identical with defined user group. In this case the role represents also to some extent the context of learning, which is widely defined by characteristics of the learner and the profile associated with this specific learner group (cf. Kritzenberger/Herczeg 2000).

Tools: support for the execution of tasks. In the example of educational systems this can be represented by a teaching or pedagogic model. (The leading question: Which method fits best for reaching specific teaching goals?)

The framework described above is realized in a prototype (cf. Figure 3).

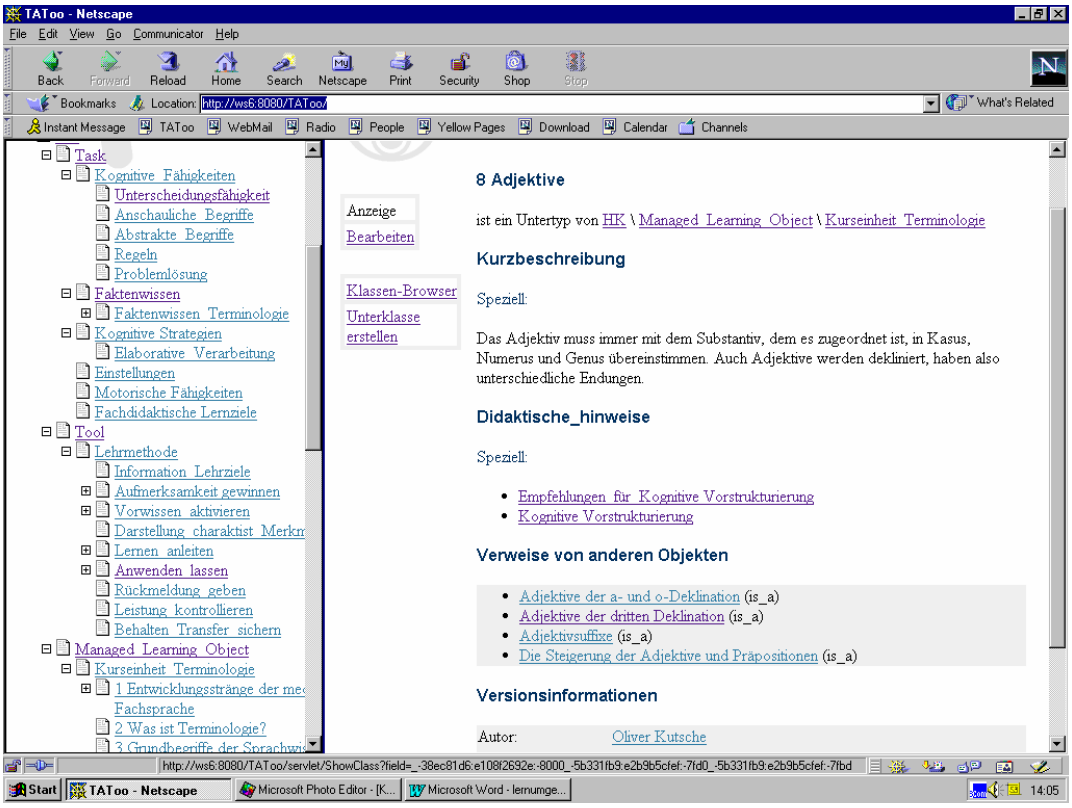

Figure 3: Screenshot of prototype modelling a course unit on medical terminology

Figure 3 shows a screenshot of the prototype modelling a part of the course on medical terminology (subject: adjectives) as part of a course of studies in medical computer science. In the left frame there are the elements from the framework with task, tool, managed learning object and user group. In the right frame there is a view on the managed learning object "Adjectives". The screenshot shows an under the heading "Verweise von anderen Objekten" how the content authors had considered an ordering and logical sequencing of the content. Furthermore there is general didactic knowledge on how to structure courses, which should also be considered when the course unit is constructed. This information needs to be stated explicitly, if it should be present in all stages of development and forms a kind of meta information if the learning objects should be re-used with different views on the material, e.g. for building a course for another user group.

4 Enhancing of OOA with Scenarios

As discussed in the previous section, data are collected, specified and stored according to the described framework. As the framework allows for further attributes describing parameters of the context of use or user characteristics, it is possible to add information, which is essential for the usability of the system. Usability relevant information is for example characteristics and needs of special learner groups. These data provide specific information, which can be gathered by data collection in organizations by asking questions like who are potential users, or by analysis of imagined scenarios (cf. Carroll 1995).

Lets make an example from the domain of educational material to get things more clear. For the use of educational material it is normally important to be aware of the previous knowledge of the users, as the learning material has to be adapted accordingly. Although this seems to be a very clear parameter for system variation, these data are less precise than they seem to be. For example, it is not clear right from the beginning, in what way the educational modules have to be altered, because usability information like previous knowledge is not sufficiently defined. Nevertheless, it is important to assess these data and possibly refine this information in the course of the development process unless it is precise enough to make relevant design decisions. If it is integrated into the object-oriented framework, it can be handled with object-oriented methods, as demonstrated with the following example.

An example scenario: Learner groups in a virtual university

The educational material offered in a virtual university will not only be used by traditional students of universities but by a variety of different user. Therefore, one can distinguish different user groups, which can be further distinguished by their basic needs according to the available educational material. For example, learner group 1 wants to do basic studies in the domain in order to pass exams and earn diploma. The learners want to study all modules relevant to the curriculum, need guided tours to differ relevant from irrelevant knowledge with respect to passing exams. Learner group 2 wants to do post-graduate studies. They have already passed an exam and have learnt how to learn. Learner group 3 does not want to go through all the knowledge modules offered in the database, but wants to study only some modules for task-oriented knowledge acquisition. Often a current problem or a knowledge gap is the starting point for learning. Learner group 4 is characterized by exploring the knowledge domain according to personal interests. The behavior of learners in this group is comparable to browsing libraries, reading books and magazines.

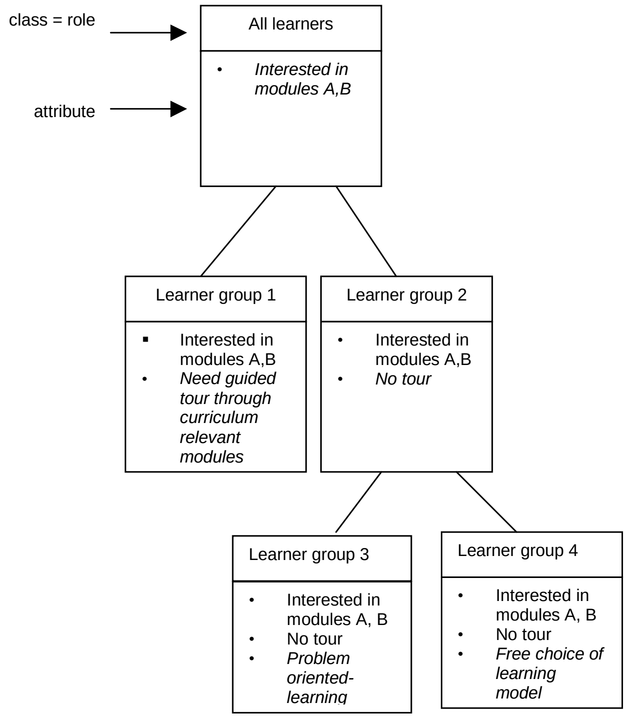

Figure 4: Hierarchy of learner groups demonstrating attribute inheritance for usability information

The data are stored as XML data records in a relational data base (cf. Kutsche 2000). An important advantage of this method is that classes of information are introduced. Such classes are for example, "user attributes", "content" and so on. These classes can be freely combined into views on the database, like the example of a role dependent view in figure 5 illustrates. In general, modeling and storing of classes with varying attributes allow flexibility in storing and selection of data. To take again the example of the learning environment, this data organization allows to cover attributes, which are valid for the whole context, and attributes, which are valid only for certain parts of the context. Moreover, another advantage is object-orientation, which allows generalization or abstraction with different detailing of information at different abstraction levels. So, developers don't have to deal with too many irrelevant data and information. Even numerous and complex attributes stay manageable, because specific attributes are only valid for certain user groups.

Another helpful quality of XML-based relational databases for the development process is, that it can be easily translated into other SGML-type languages like XHTML or LaTeX in order to visualize and document the contents. As the framework is web-based with server pages and the database is capable of serving a network, it is available for each of the members of the design team to insert data directly and can be used even for distributed teams.

5 Conclusions

The object-oriented framework described above allows flexible management of usability data during the whole development process. Even complex and informal data, e.g. from scenario descriptions, can be included and selected according to situational and role specific needs of users. As far as our experience with the application of hypermedia learning environments is concerned, the framework offers the possibility to include learning modules and connect usability information on different levels of abstraction. For the our application of a learning environment this will also be a basis for adaptation criteria.

For the future work, the framework for modeling learning environments will be further detailed. As software engineering is an iterative process that requires constant updating, the usefulness of our framework will be further proved by practice.

References

1. Brusilovsky, P. (1998): Methods and Techniques for Adaptive Hypermedia. In: Brusilovsky, P.; Kobsa, A.; Vassileva, J. (Eds.): Adaptive Hypertext and Hypermedia. Dordrecht, Boston, London: Kluwer Academic Publishers, pp. 1-43
2. Carroll, J.M. (Ed.): (1995): Scenario-Based Design. Envisioning Work and Technology in System Development. New York: Wiley
3. De la Teja, I.; Longpré, A.; Paquette, G. (2000): Designing Adaptable Learning Environments for the Web: A Case Study. In: Proceedings of ED-MEDIA 2000. World Conference on Educational Hypermedia, Multimedia and Telecommunications. 26th June – 1st July 2000. Montréal, Canada. AACE: Association for the Advancement of Computing in Education, pp. 243-248
4. Hartwig, R.; Kritzenberger, H.; Herczeg, M. (2000): Course Production Applying Object Oriented Software Engineering Techniques. In: Proceedings of ED-MEDIA 2000. World Conference on Educational Hypermedia, Multimedia and Telecommunications. 26th June – 1st July 2000, Montréal, Canada. AACE: Association for the Advancement of Computing in Education, pp. 1627-1628
5. Herczeg, M. (1999): A Task Analysis Framework for Management Systems and Decision Support Systems. In: Proceedings of AoM/IaoM. 17. International Conference on Computer Science. San Diego, California, 6th – 8th August 1999, pp. 29-34
6. Herczeg, M. (2001, to appear): A Task Analysis and Design Framework for Management Systems and Decision Support Systems. To appear in: International Journal of Intelligent Control Systems
7. Holtzblatt, K.; Beyer, H. (1993): Making Customer-Centered Design Work for Teams. Communications of the ACM 10, pp. 93-101
8. ISO 13407: 1999 Human-centered design process for interactive systems
9. Kritzenberger, H.; Herczeg, M. (2000): Completing Design Concepts for Lifelong Learning. In: Proceedings of ED-MEDIA 2000. World Conference on Educational Hypermedia, Multimedia and Telecommunications. 26th June – 1st July 2000, Montréal, Canada. AACE: Association for the Advancement of Computing in Education, pp. 1374-1375
10. Kutsche, Oliver (2000): Proof-of-concept der datenbank- und web-basierten Unterstützung von Entwicklungsprozessen für einen Prototypen. Studienarbeit, Informatik, Medizinische Universität zu Lübeck, August 2000