In *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Proceedings of the* 6th Workkshop, ABIS-98, U. Timm and M. Roessel (Eds.), 1998.

Towards a Methodology for Adaptive Hypermedia Systems Development

Nora Koch

Ludwig-Maximilians-Universität München
Institut für Informatik
Oettingenstr. 67
D-80538 München

Tel. +49 89 2178 2177
Fax +49 89 2178 2152
kochn@informatik.uni-muenchen.de

Abstract

Adaptive hypermedia (AH) systems that adapt themselves to the user's characteristics generally include for this purpose a user model component. The designer has to specify different solutions for content, navigation and presentation, and define adaptation criteria. To support the development of AH applications an engineering approach is required that considers user modelling and adaptive aspects. This paper presents a Methodology for Adaptive Hypermedia Systems Development (AHDM) covering the whole life cycle of AH applications. It includes phases for the development of the user model, the adaptive interface and the dialogue component responsible for the modelling of the user's behaviour. For each analysis and design phase an appropriate notation is proposed.

Keywords
adaptive hypermedia systems, user modelling, hypermedia design method, web engineering.

1 Introduction

When developing large and complex systems it is important - as stressed by Jacobson [Jacobs92] - to be guided by a sequence of steps to be performed, to know how the different steps co-operate and how they fit into the development process as a whole. Adaptive hypermedia applications are large and almost always complex user-model-based hypermedia systems and consequently, not an exception. Developers of adaptive hypermedia systems require special support in the design of the adaptable interface and structure as well as in defining the user model. Guidelines to support the development process of adaptive hypermedia applications are lacking. Thus, these large interlinked applications, usually based on user models, are difficult to design and hard to maintain.
This paper presents a brief description of a first engineering approach covering the whole life cycle in the development of user-model-based adaptive hypermedia systems (AHS), i.e. a systematic and methodological procedure that describes the steps to follow in the development process of AHS.
The Adaptive Hypermedia Systems Development Methodology (AHDM), we defined, is an object-oriented approach based on the classical phases in software systems production comprising analysis, design, implementation and tests. Additionally, it takes aspects into account that are important in the acquisition process of a software system, which includes planning, risk management and control. This methodology therefore, starts with a feasibility

study of the application development and, as a further step, includes the planning of the strategy to be followed in the design and implementation.

The test phase is not the final step in the development process with the purpose to check the functionality of the application, e.g. no dangling links. Quality control activities are planned in our methodology along the whole process, i.e. validation of the analysis specification and verification of the implementation results against the requirements.

The development of adaptive systems requires the specification of the domain, user and dialogue models as exposed in [BenMuy93, McTear93]. Thus, we add special steps for the analysis and design of the user model and for the dialogue model that records the user's behaviour. Special support is required for the adaptive interface and adaptive navigation design and implementation. In each phase AHDM reuses, whenever possible, models, techniques and notations that have been successfully proven in software engineering processes, such as the standard notation UML [UML97], Use Cases [Jacobs92] or Statecharts [Harel87].

In recent years several methods have been developed for the design of hypermedia applications: OOHDM [ScRoBa96], RMM [IsStBa95], EORM [Lange95], WSDM [TroLeu98]. All of them describe a sequence of steps for the design process; the approach is either object-oriented or based on the entity-relation model. Another methodology is proposed by Olsina: the HFPM strategy that suggests a list of tasks to be performed when developing hypermedia applications [Olsina98].

None of these methods consider neither user modelling nor user-model-based adaptation. Our methodology presents similarities with these methods such as the clear separation of content, navigation and presentation design. In contrast to the other methods we include user model design and adaptive mechanisms as a central part of our methodology. In addition, AHDM describes a web engineering process covering all phases from the feasibility study to the maintenance.

Section 2 outlines the Methodology for Adaptive Hypermedia Systems Development. Sections 3 to 5 describe the first three phases of the method and the appropriate software engineering techniques that can be used for the analysis and modelling activities. In Section 6 we present some concluding remarks and an overview of the future work.

2 A Development Process for Adaptive Hypermedia Systems

From a software-engineering point of view the development process of AHS comprises the usual phases: analysis, design, implementation, test and maintenance. From the procurement process point of view the acquisition phases are goal definition, strategy planning, contracting and control. There are many common aspects in both processes such as analysis of the problem requirements, definition of milestones for the design and implementation, and definition of quality of the deliverables. The idea is to enrich the software development process with aspects from the procurement process.

We hereby define the Methodology for Adaptive Hypermedia Systems Development (AHDM) which comprise the following phases:

- Feasibility study
- Analysis
    - ⇒ User analysis
    - ⇒ Requirement analysis
    - ⇒ Strategy planning

- Design
    - ⇒ User model design
    - ⇒ Conceptual modelling
    - ⇒ Navigational design
    - ⇒ Abstract interface design
    - ⇒ Dialogue modelling
- Implementation
- Quality tests
- Maintenance

These steps of AHDM describe the life cycle of AHS and are performed in a mix of incremental, iterative and prototype-based development style. Implementation, quality assurance and maintenance are not within the scope of this paper.

Case study: In the following we illustrate the development process by an exercises session of the SmexWeb System [SmexWeb98, Albrec98, Tiller98] which presents EBNF exercises for an Introductory Course in Computer Science.
SmexWeb (Student Modelled Exercising on the Web) was developed at the Ludwig-Maximilians University München. It is an adaptive web-based tutoring system which implements a user model considering cognitive and knowledge aspects as well as general abilities of the students. Adaptation is performed through annotated, ordered and hidden links as well as adaptive presentation and passive navigation.

3 Feasibility study

The *idea* is the essential element of the feasibility study. The process starts with the conception. The goal is to develop and evaluate the idea for an AHS, i.e. the need for an adaptive capability and benefits in developing an application based on it. Therefore, the global functional requirements of the AH application have to be defined, and a first budget and schedule plan to be worked out. Costs of the project must be estimated in this phase, because this is a crucial element in determining the feasibility of the project.
It may be advisable to develop a prototype, but one has to consider the effort that must be minimised just in case the idea fails to live up. Depending of the results of this study it can be decided whether it is worthwhile to develop an hypermedia application which dynamically adapts to the user profile.
Techniques supporting feasibility study vary from informal *textual description*, through *checklists* and *spread sheets* to implementation-oriented ones like paper or computer based *storyboarding* as proposed by Boyle [Boyle97]. Results from *prototyping* are a meaningful but expensive alternative.
The idea of the exercises session of the case study was to offer exercises and definitions about the chosen subject to a very heterogeneous group of students (different background, ages from 20 to 70, etc.). According to their profiles the exercises are presented with different degrees of difficulties, described in a formal or pragmatic way, including or not examples, etc.

4 Analysis

The objectives of this process is to analyse and specify the AH application that is to be built. This phase as well as the following ones will only be realised in case the feasibility study was positive.
This process specifies what the AHS will do (system behaviour), why it will do it (function of the system), and how it is planned (system implementation). In addition, we have to determine what the user will do with the AH application (required system's functionality), why they will do it (user's goal and characteristics) and how they plan to interact (system-user interaction).
The analysis phase of AHDM comprises the following steps: user analysis, requirements analysis and strategy definition.

4.1 User Analysis

Knowledge about the potential users is of crucial importance to design the user model and the adaptive application. This is achieved by identifying user characteristics such as tasks, needs, preferences, interests, goals and knowledge about the domain. The aim of this step is to obtain the information needed to build an appropriate user model for the AHS. Therefore, it has to be determined how the user's behaviour is captured by the system and how the user model is to be adapted dynamically to this behaviour.
The fundamental user analysis activity is to carry out *structured interviews* based on checklists [KocTur97, KocSch97]. This interviewing process is divided in the successive steps of preparing questionnaires based on predefined checklists, the interview itself, and the editing of the results. The goal is to define user groups and for each group to identify the information they are interested in, their preferences and the typical navigation activities they will perform.

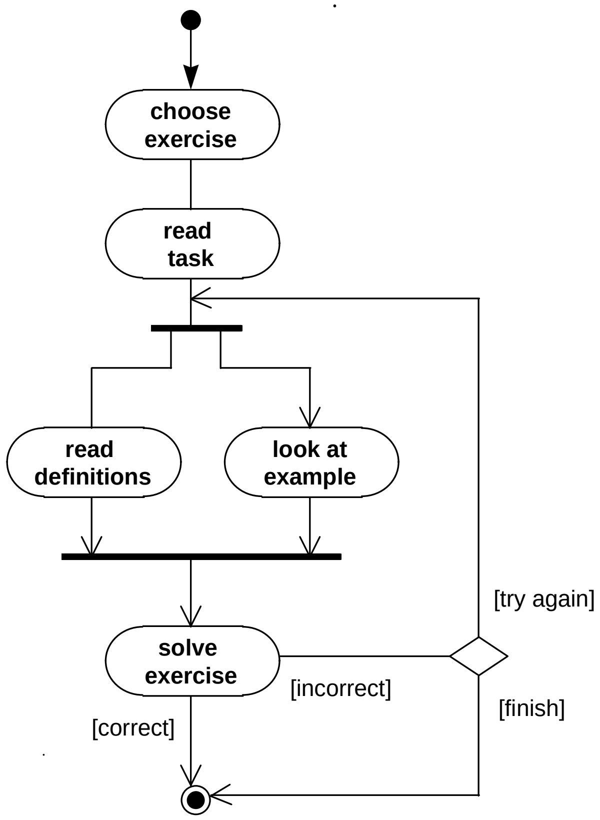

Figure 1: Typical user activities for solving an exercise

It is possible to represent graphically these sequences of activities, e.g. in UML *activity diagrams*.
For example in the SmexWeb case study we identified different types of students through an interview process: students with and without computer or web experience, with or without knowledge of programming languages or EBNF, preferences for formal or more pragmatic explanations, etc. Fig. 1 details a typical sequence of user activities for solving an exercise.

4.2 Requirement Analysis

The requirements analysis consists of a specification of the AHS objectives and a specification of the application behaviour. The functional requirements of the hypermedia application as well as of the user model have to be defined. Benyon and Murray [BenMuy93] recommend to take into account the following points in the identification of the requirements of an adaptive hypermedia system:

- defining the scope of the adaptive mechanism and methods for measuring the scope,
- how data can be obtained for the user model (interviewing or capturing of user actions), and
- how user's profile can be inferred from the dialogue record.

The requirement model can be developed using Jacobson´s *Use Cases* [Jacobs92,UML97]. Actors playing a role in the AHS have to be identified first. To define the use cases, one has to investigate what each actor wants to do with the system. An additional textual description of the use cases defining pre-conditions and post-conditions permits a more detailed specification of the requirements.

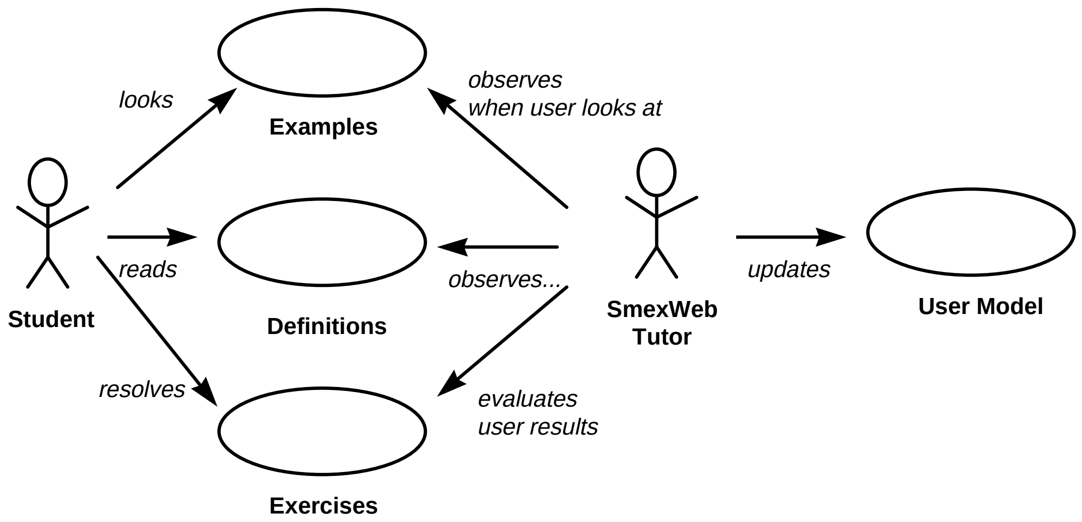

Figure 2: Use case "exercises session" with SmexWeb

SmexWeb assigns a tutor to each student that registers at the system. Thus, we can identify two actors: the student and the SmexWeb-Tutor. In Fig. 2 the use case "exercises session" for our case study is shown.

4.3 Strategy definition

A *strategy plan* for the activities is produced based on the goal definition, the requirements and the results of the user analysis. The objectives must be mapped onto a schedule, milestones have to be set and measurements need to be selected in order to achieve the goal. Costs and a schedule for milestones and deliverables must be defined for each design, implementation and test steps. The definition of both the initial and final states, problem factors, actions, milestones and deliverables will help to produce a strategy plan and a delivery plan. Initial and final states can be compared with the aim of obtaining a clear definition of the content, structure, layout and adaptive features. Typical initial states are a non-adaptive hypermedia system or a non computer-based system.

This is the starting point for the strategy definition which is done within a risk management framework that allows the analysis of the critical success factors and the actions that reduce risks. Special attention in the definition of the strategy must be paid to the milestones to be included in the design and implementation of the user model of the AHS.

The Euromethod framework [Eurome96,Wierin96] gives very general guidelines for planning and decision making. Siegel proposes in [Siegel97] very pragmatic rules for web applications development as defining the mission statement, marketing goals, competition analysis and branding. Each one of them has the same objective of producing an appropriate *strategy and delivery plan***.**

For the exercises session a plan with the sequence of examples and exercises to be designed and implemented are defined, which are determined according to the result of the two previous steps. Adaptive aspects have been analysed too. The exercises are: recognise correct EBNF sentences, build correct EBNF sentences, recognise EBNF grammar rules, build EBNF grammar rules and an application to apply the acquired knowledge.

5 Design

The design phase of AHDM consists of a model-based approach for building adaptive hypermedia applications. It comprises user model design, conceptual modelling, navigational design, user interface design and dialogue modelling. This provides a clear separation of the information the user can access, how this information is structured and how it is presented to the user.

AHDM is a user-centred design approach that takes into account user characteristics for the model of each phase. Content and navigation is modelled from the viewpoint of the user types that have been identified.

In the design of AHS some special aspects related to content, navigation and presentation have to be considered. Alternative content for the same theme makes it possible to show to each user an appropriate version of the theme. The same information can have different layouts for different users. Different navigation paths result from adaptive navigation. Consequently, some information or some nodes may be visible for some users but not for others.

5.1 User Model Design

The user model design step aims at building a user model that represents knowledge, goals and/or individual features, such as preferences, interests and tasks of the users. The model is the view the system has of the user. Its main purpose is to influence interface generation. The static aspects of the user model are described with an object-oriented domain model.

For the dynamic aspects AHDM proposes the definition of *aspects, states and transitions.* Each aspect represents one characteristic (attribute) to be modelled or a combination of user characteristics. States are the different possible values for each aspect. Transitions are defined by a pre-condition, an event and a post-condition (indicated with letters *a* to *g* in Fig.3).
Results of this step are a set of definitions of the user attributes and their states and relations (transitions) and a graphical representation with *user charts.*
State diagrams of UML (based on Harel´s Statecharts [Harel87]) have been chosen for the diagrammatic notation of the user charts. Each user chart shows the different states an aspect may go through during its life in response to some stimuli, together with its response and actions. An aspect remains in a state for a finite time. An initial state or initial transition must be defined. Some events may be related, i.e. messages may be send from one user chart to another. Three related user charts are shown in Fig. 3 corresponding to a partial diagram of the user model of our case study.

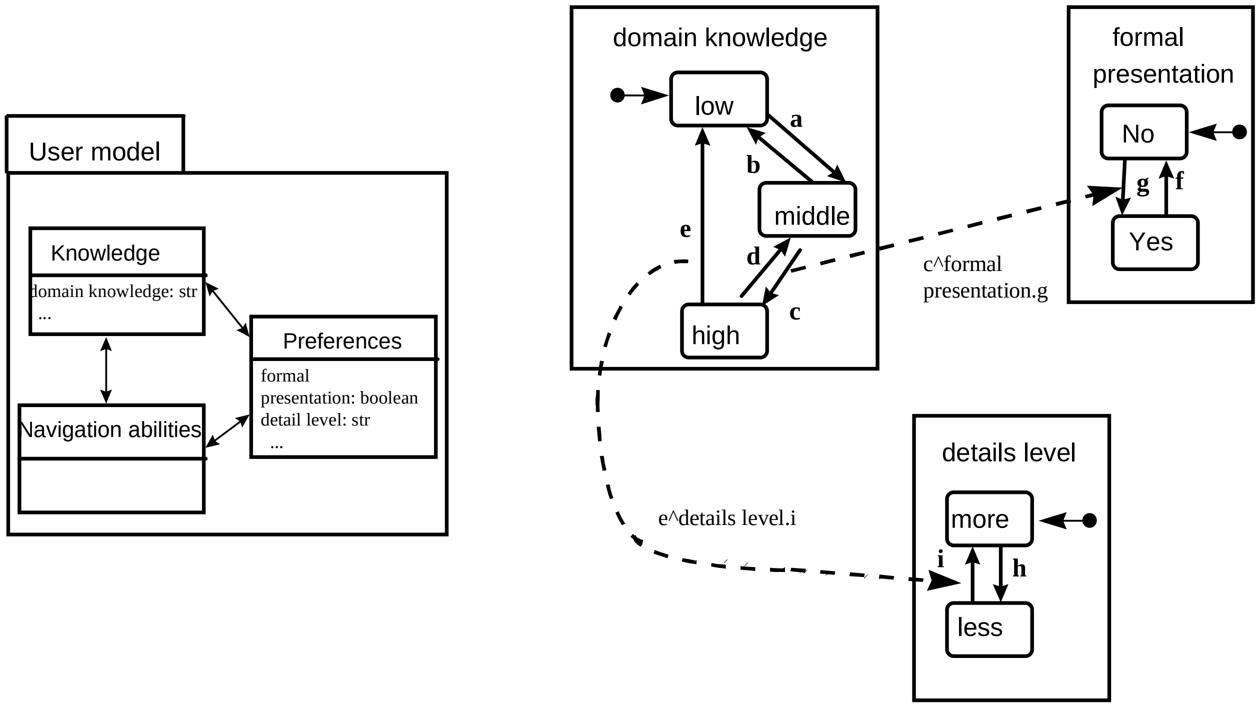

Figure 3: Partial representation of the user model

5.2 Conceptual Modelling

The conceptual design aims to build a domain model including all the concepts that are relevant to the application and the different users or user groups identified in the previous steps. The main objective is to capture the domain semantics with as little concern as possible of the navigation paths, presentation and interaction aspects. Decisions as whether each concept corresponds to one hypermedia page, a hypermedia document or if the page is generated on-the-fly based on the frame-based internal representation of domain concepts, are postponed to the implementation phase.
A *set of definitions* of sub-systems, classes, objects and relations constitute the first resulting products of this step. They can be summarised in a *conceptual schema.* The UML class

diagram and package notation [UML97] has been chosen for the diagrammatic representation of the conceptual schema of our example; OMT as in [ScRoBa96] or ER-modelling as in [IsStBa95] could have been selected as well.
Fig. 4 shows the class diagram for the conceptual schema of the exercises session. Here we have already considered that the content will be adaptive depending on the user model. For example several help texts are associated with a particular exercise.

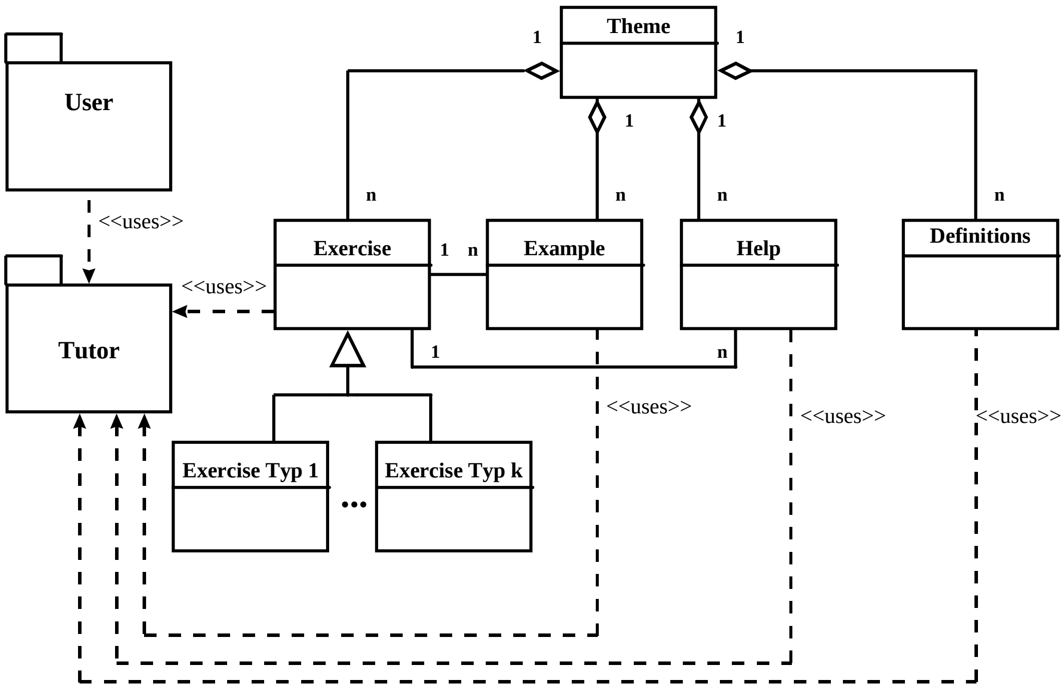

Figure 4: Class diagram of the exercises session.

5.3 Navigational Design

Navigational design is a critical step in the design of every hypermedia application. The different views that are relevant for the different users are derived from the conceptual model. These describe the different navigational possibilities that the application will offer to the users, i.e. define the structure of the hyperspace.
Navigation is defined in navigational classes, links, anchors, contexts and access structures. AHDM uses the concept of navigational *context* introduced by OOHDM [SchRos98]. A navigational context is a set of nodes, links, context classes and other nested navigational contexts. Context classes indicate which information is shown and which anchors are available when accessing the object in a particular context. Navigational contexts are triggered by *access structures* - indexes, queries, menus.
These model objects are used in the three different diagrams: *navigational class schema*, *context schema* and *navigational charts*. The first one is used to specify navigation between classes; the second one details the navigation path using contexts and access structures; finally, the navigational charts visualise dynamic aspects, like for example which objects are opened or closed after a mouse click.
Different notations are used for these diagrams: Navigational class schemes are represented using UML notation for class diagrams. For the contextual schema a specially notation

proposed by [ScRoBa96] can be used. Navigational charts are based on UML state diagrams [UML97]. Fig. 5 shows a simplified context schema for our case study.

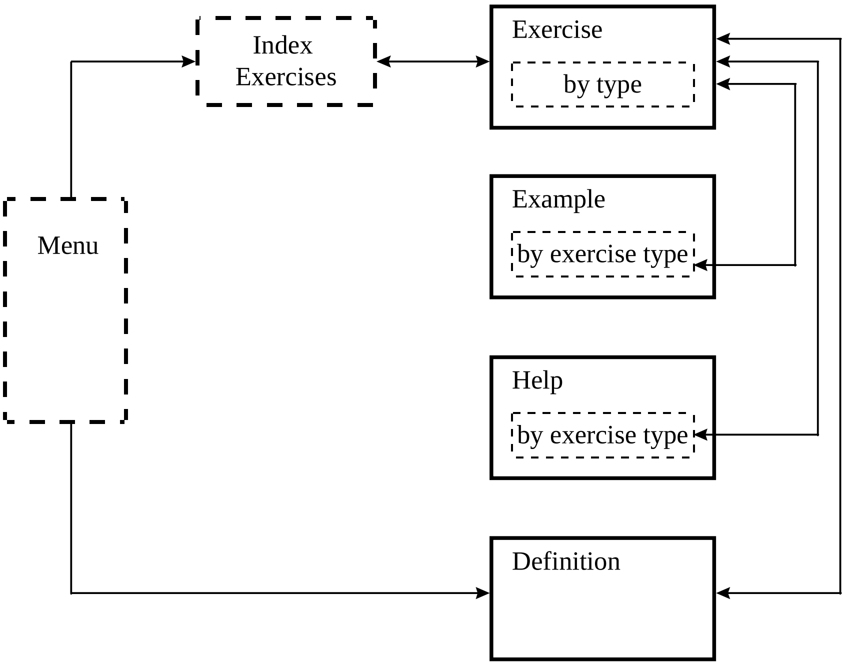

Figure 5: Context schema for the exercises session

5.4 Abstract Interface Design

Interface design means defining the way in which navigational objects will appear, which interface objects will activate navigation, and which interface transformations will take place. The developer has to decide in this design step which type of adaptation is chosen for the AH application.

In [Brusil96,EkBrSc98] Brusilovsky distinguishes two types of hypermedia adaptation: adaptive presentation (content-level adaptation) and adaptive navigation support (link-level adaptation). The first present to the user content adapted to the user's profile. The second one is suggesting through different adaptive techniques the "best" link, sometimes forcing him to follow a determined path. The most popular techniques for adaptive navigation are: direct guidance, adaptive ordering, adaptive hiding and adaptive annotation. In the SmexWeb exercises session for example, apart from adaptive presentation three of these techniques are used in combination, i.e. annotation, ordering and hiding of links.

The *Abstract Data Views* (ADVs) approach design [CaCoLu93] can be used (as it is in OOHDM) for describing the user interface of the hypermedia application. ADVs are objects, which represent only the interface and the state, not the implementation. This interface can be exercised through messages (in particular external events generated by the user). Aggregation is used to define interface perceivable objects as composition of lower-level ADVs. Inheritance mechanisms provide a framework for defining hierarchies of interface objects.

Dynamic and static aspects of the abstract interfaces are graphically presented in configuration diagrams and ADV charts. These generalisation/specialisation hierarchies in ADVs make a graphical representation of all interface objects possible. However, we consider that for frequently used interface objects, such as anchors, buttons or lists, a special diagrammatic

representation is required. Therefore, our approach distinguishes with a special notation input, selection, interaction and navigation model elements.
Additionally, AHDM extends the ADV model with model objects for content-level and link-level adaptation. Anchors can be hidden or annotated and a group of anchors can be ordered. The resulting notation permits a clear specification of contents (adaptive presentation) and links (adaptive navigation) changing dynamically according to user model states.
Fig. 6 shows the configuration diagram for the ADV exercise of our case study. Notation for interactive elements (underlined text) can be seen in the navigation bar at the bottom and the links to other exercises on the right. Double-lined boxes are used for adaptive presentation. Anchors to type of exercises 2 and 3 are annotated (indicated with *) and ordered (therefore included in a dashed-box). The anchor to exercise type 4 may be hidden. The dashed-lines indicate navigation within the same ADV.

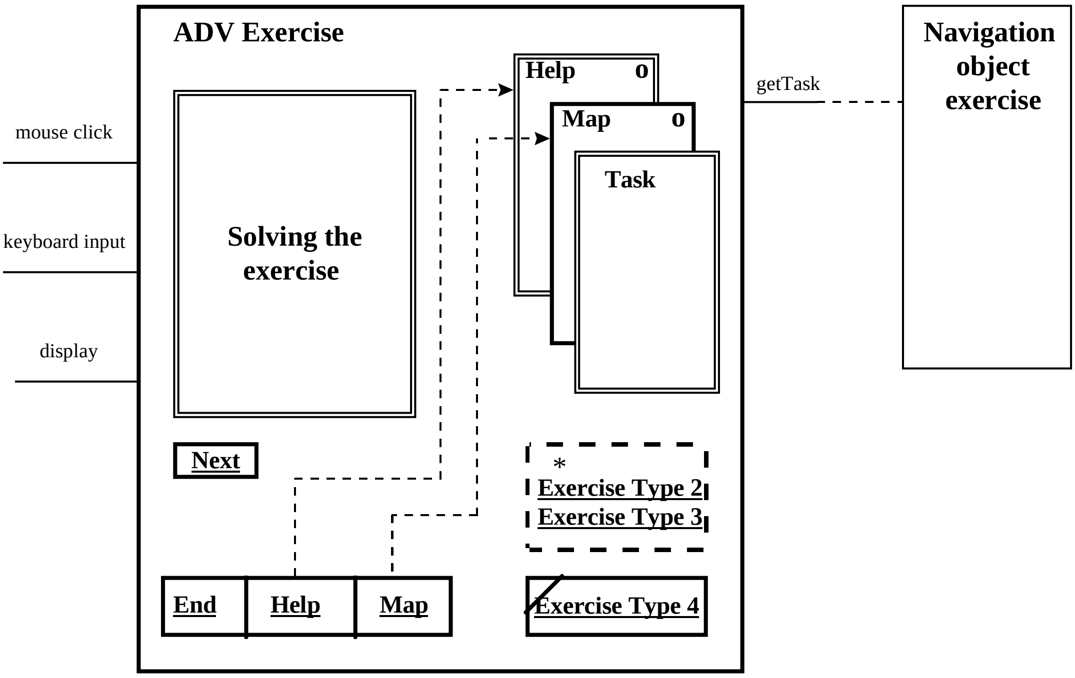

Figure 6: Configuration diagram for an exercises session

5.5 Dialogue modelling

To obtain relevant information of the user characteristics, the user behaviour must be observed and registered in a dialogue model. We call these dialogue model elements interactions as in [BenMur93]. From the data gathered, the system can make inferences about the user's belief, plan, knowledge, etc. The user model then is changing dynamically according to this dialogue model defined by a set of dialogue rules [MurMcT97]. Finally, adaptive navigation and adaptive presentation will occur based on the actual state of the user model.
There are three main aspects for specifying the dialogue model:

- *capturing the appropriate data*. The raw data is obtained maintaining a dialogue history which records all interactions for a particular user. This interactions may be: keystrokes, mouse clicks or movements, time elapsed between interactions or system messages.
- *selecting the relevant information.* Only certain interactions produced under certain circumstances contribute with information to infer the user's profile.

- *inferring from the dialogue rules***.** With the selected data and a set of defined dialogue rules a cognitive interpretation is assigned to the user interaction or a sequence of interactions.

Thus, we distinguish three types of dialogue rules:

- *information-based rules* uses information obtained from *direct input* of the user, such as by answers from an initial interview or from *tests or exercises* about domain knowledge.
- *behaviour-based rules* are classified as *time-based* or *link-based*. Time-based rules take into account the elapsed time between two user actions; link-based rules are defined by user actions, such as mouse clicks on anchors.
- *history-based rules* are defined based on sequences of user mouse actions, keyboard inputs or periods of inactivity.

6 Conclusions and future work

Adaptive hypermedia developments have been produced until now without using any strategy or systematic, well-defined process. We have presented AHDM, an engineering approach for the development of adaptive hypermedia systems. In this paper the first three phases: feasibility study, analysis and design have been briefly explained. Special focus has been put on user and dialogue modelling as well as on the adaptive features of AHS. Appropriate techniques that support the activities of each step have been outlined.

We hope that this contribution will convince developers of hypermedia applications to take full advantage of the benefits of dynamic adaptation in their systems. Implementation of user-model-based hypermedia applications will introduce flexibility in hypermedia applications which are used by different classes of users reducing the "lost in hyperspace" problem of classic hypermedia systems by providing intelligent guidance.

Our future work will concentrate its attention on refining the techniques and notations. In particular, UML or an extension of UML will be compared with the ADV modelling approach for the abstract interface design. A proposal for a simplified navigational design will also be devised.

References

[Albrec98] F. Albrecht, *SmexWeb: Ein adaptives web-basiertes Übungssystem.* Diplomarbeit, Ludwig-Maximilians-Universität München, 1998.

[BenMuy93] D. Benyon, D. Murray, *Applying User Modeling to Human-Computer Interaction Design.* In Artificial Intelligence Review, Kluwer Academic Publishers, 6, 43-49, 1993.

[Boyle97] T. Boyle, *Design for Multimedia Learning.* Prentice Hall, 1997.

[Brusil96] P. Brusilovsky, *Adaptive Hypermedia: An Attemp to Analyze and Generalize.* In Multimedia, Hypermedia and Virtual Reality, Lecture Notes in Computer Science 1077, Eds. P. Brusilovsky, P. Kommers, N. Streitz, Springer Verlag, 288-304, 1996.

[CaCoLu93] L. Carneiro, D. Cowan, C. Lucena, *Introducing ADVcharts: a Graphical Specification of Abstract Data Views.* In Proceedings of CASCON´93, 1993.

[EkBrSc98] J. Eklund, P. Brusilovsky , E. Schwarz, *A Study of Adaptive Link Annotation in Educational Hypermedia.* In Proceedings of EDMEDIA´98, 1998.

[Eurome96] Euromethod, 1996, http://www.fast.de/Euromethod/index.html.

[Jacobs92] I. Jacobson, *Object-Oriented Software Engineering: A Use Case Driven Approach.* ACM Press, 1992.

[Harel87] D. Harel, *Statecharts: A Visual Formalism for Complex Systems.* Science of Computer Programming, 8(3), 1987.

[IsStBa95] T. Isakowitz, E. Stohr, and P. Balasubramanian, *A Methodology for the Design of Structured Hypermedia Applications*. Communications of the ACM, 38(8),34-44, 1995.

[KocTur97] N. Koch, and A. Turk, *Towards a Methodical Development of ElectronicCcatalogues.* In International Journal of Electronic Markets, University of St. Gallen, Vol. 7(3), 28-31, 1997.

[KocSch97] N. Koch, and J. Schneeberger, *Integrated Assistance for the Development of Electronic Product Catalogues.* In Proceedings of Symposium of Software Technology (SoST´97), 101-112, 1997.

[Lange95] D. Lange, *An Object-Oriented Design Approach for Developing Hypermedia Information System*s. Research Report RT0112, IBM Research, Tokyo Research Laboratory, Japan, 1995.

[McTear93] M. McTear, *User modelling for Adaptive Computer Systems: A Survey of Recent Developments.* Artificial Intelligence Review, Vol. 7, Kluwer Academic Publishers, 1993.

[MurMcT97] M. Murphy, M. McTear, *Learner Modelling for Intelligent CALL.* In Proceeding of Sixth International User Modelling Conference, Springer Verlag, 301-312, 1997.

[Olsina98] L. Olsina, *Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy*. Presented at First International Workshop on Hypermedia Development, Hypertext´98, 1998.

[ScRoBa96] D. Schwabe, G. Rossi, S. Barbosa, *Systematic Hypermedia Design with OOHDM.* In Proceedings of the ACM International Conference on Hypertext (Hypertext´96), 1996.

[Siegel97] D. Siegel, *Secrets of Successful Web Sites*. Macmillan Computer Publishing, 1997.

[SmexWeb98] SmexWeb: Student Modelling Exercising on the Web, http://pst1.pst.informatik.uni-muenchen.de:8000/

[Tiller98] T. Tiller, *Eine Hypertext-Struktur für abgeschlossene adaptive Systeme.* Diplomarbeit. Ludwig-Maximilians-Universität München, 1998.

[TroLeu98] O. de Troyer, C. Leune, *WSDM: A User Centered Design Method for Web Sites*. In Proceedings of WWW7 Conference, 1998.

[UML97] UML Document Set, Version 1.1 September 1997, Rational, http://www.rational.com/uml/references

[Wierin96] R. Wieringa, *Requirements Engineering: Framework for Understanding.* John Wiley & Sons, 1996.