Hypermedia Modelling Using UML

Peter Dolog *
dolog@dcs.elf.stuba.sk

Mária Bieliková*
bielik@elf.stuba.sk

Abstract: This paper discusses an approach to hypermedia modelling using the Unified Modelling Language (UML). We describe hypermedia models based on four established views of hypermedia system: application domain model, navigation model, presentation model and user model. The approach emphasizes UML state diagrams as capable technique for navigation modelling. We relate our approach to other types of models. Adaptation modelling according to the user model is also presented.

Keywords: hypermedia modelling, behavioural model, UML, personalised hypermedia.

1 Introduction

The paper presents an approach to hypermedia application modelling using the Unified Modelling Language (UML). Hypermedia application is considered as a system, which provides information related to particular domain. Presented information is divided into chunks interconnected by active links. To provide comprehensive and consistent model of a hypermedia application, several modelling perspectives were established [5]. Principal perspective is concerned with characteristics of solution (hypermedia) environment, i.e., "what is modelled in hypermedia application" question is resolved. According to this point of view we distinguish:

- *application domain model:* incorporates concepts with their mutual relationships in particular domain. This model determines, which information will be provided by hypermedia application;
- *navigation model:* defines a way, how application will navigate a user in the information net. Navigation model groups information chunks into contexts and interconnects them by links;
- *presentation model:* is concerned with appearance of information chunks to a user. It also defines spatial layout and content of information chunks related to the user interface. Presentation model defines presentation classes or objects, spatial relationships between them and content associated with them.

To support personalisation a *user model* should be incorporated. The user model determines user roles and features, which are important for adaptation and personalisation.

Mentioned views are gradually refined from requirements to implementation.

* Department of Computer Science and Engineering, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia

Several approaches to hypermedia modelling appear in the literature. Most of them consider all mentioned views in modelling. Differences are primarily in modelling techniques and corresponding notations used together with aspect (structural or behavioural) considered in the process of hypermedia application modelling.

Hypermedia application is an interactive system where reactive behaviour of the application is important especially in navigation and presentation views. Existing approaches employ largely structural models and use E-R diagrams (e.g., Hypermedia Design Method (HDM) [13], Relationship Management Methodology (RMM) [17], Web Modelling Language (WebML) [2]), or class diagrams (e.g., UHML (extended UML by Nora Koch) [15], W2000 (successor of HDM) [1], Object-Oriented HDM (OOHDM) [18]). Behavioural techniques such as petri nets or state diagrams are underestimated and are not widely used. Petri net based model is defined for example in [12]. However, we did not find any design method, which includes it. OOHDM [18] mentions navigation transformation model, which is expressed by Harel state diagrams. It is very similar to Hypermedia Model Based on Statecharts Method (HMBS/M) [4] where both contents and navigation instance schema are modelled by a state diagram.

We proposed UML statechart based approach to navigation modelling in [6] and adaptive navigation modelling in [7]. We believe that behavioural models are crucial for hypermedia application modelling in all its aspects (not only navigation). On the other hand, structural model is also effective vehicle for expressing particular view on the hypermedia system. The goal of this paper is to address modelling approach where behavioural statechart based navigation model plays important role besides the structural models.

The rest of the paper is organised as follows. The application domain model is described in Section 2. Section 3 is devoted to navigation modelling. In Section 4 we discuss how a user model can be expressed by the UML. Next, in Section 5, the presentation model is discussed. The paper concludes with our conclusions and some remarks for the future work.

2 Application domain modelling

All known approaches to hypermedia modelling start with some kind of application domain modelling. Main element for modelling application domain structure is a concept. Concepts are connected by relationships such as association, generalisation/specialisation, and aggregation. Concepts can have their features.

We use UML class diagram for conceptual modelling in application domain modelling. We also recommend to use activity diagrams for modelling activities in application domain. A concept is expressed in the UML by a class or an object. Features or properties are expressed as attributes or methods, operators and procedures. The basic structural relationships are generalisation/specialisation, containment or aggregation, and general association. Another mechanism to express containment is the package mechanism. Accessing or using the package element from another package is realised by the import or access dependency relationships, which are assigned to the corresponding packages.

Figure 1 shows an example of simple domain model. We took the example from the subdomain of Software Engineering — Requirements Engineering. For space limitations we restricted the concepts only to higher level of abstraction and taken into account only part of the mentioned domain.

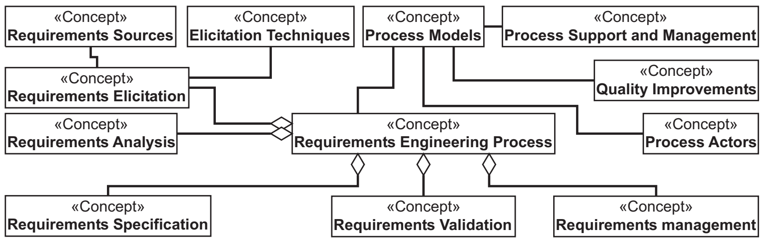

Fig. 1: Conceptual model for software engineering according to [16].

Application domain model refinements lead to a database schema for the hypermedia application. The content is then assigned directly to the database schema.

Behavioural model of an application domain can be built too. This model comprises activities related to the domain. The UML activity diagram is used for these purposes. Activity model is useful when hypermedia application should present information about some activities. This model can serve as a base for navigation, e.g., information is presented in order to activity performance.

The example of such behavioural model is activity diagram, which interconnects concepts of the *Requirements Engineering Process* (see Figure 2). The activities are taken from the conceptual view where they are expressed as concepts.

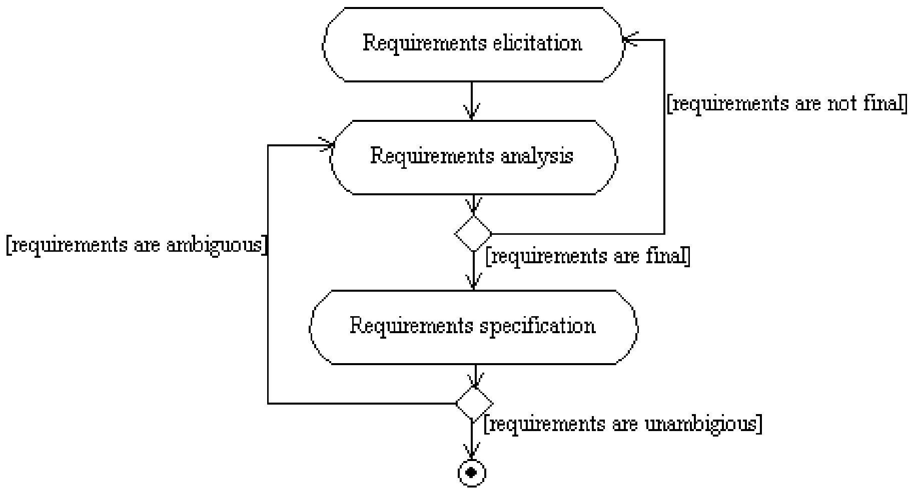

Fig. 2: Requirements engineering activity flow.

3 Navigation modelling

Navigation in hypermedia application reflects problem of a user orientation (local or global) in huge amount of information. Navigation problem (lost in hyperspace) can be reduced by appropriate composition of information chunks into information components and adequate interconnection between them. Creating a navigation model, i.e. modelling selection and grouping related information into nodes and their interconnection by links, would highly support the process of hypermedia application development. Composition of related information can be achieved by a query over the conceptual model. This requires some kind of query language over conceptual model. Another possibility is to map required part of conceptual model into the navigation model. Special type of mapping is instantiation.

Our approach is based on mapping the class diagram of conceptual model into the navigation model. Navigation model is represented by the UML state diagram. Details about this mapping can be found in [7]. State transition is raised by events, which mostly appear as a result of interaction of the user with hypermedia application. Accordingly, the first step in creating a navigation model is reasoning about high level interaction scheme. This is modelled by a sequence diagram (example can be found in [6, 7]).

Example of a state diagram is depicted in Figure 3. The guidance is illustrated in three themes: *Requirements engineering overview*, *Elicitation techniques*, and *Supporting Features*. The overview of requirements engineering contains four topics: *Introduction*, *Compositions of main activities*, *User* and *Requirements engineer* represented by states. The guidance in navigation (denoted also as browsing semantics [6]) is specified with the next link (transition). We use the history symbol (H) to indicate that the user is taken into the same state as he was before exiting the system.

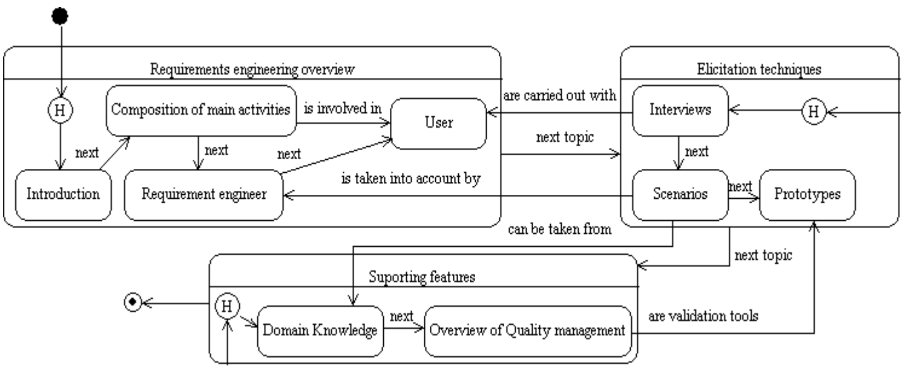

Fig. 3: An example of navigation model for requirements engineering lecture.

4 User modelling

Information is presented to different users. The users often have different goals, preferences and level of knowledge about corresponding concept or set of corresponding concepts.

The user model includes various characteristics of the user: user's goals/tasks, interests, knowledge, background, hyperspace experience, and preferences. There are several possibilities how to express these characteristics. We took as an example the user model, which is used in the adaptive hypermedia system, developed by De Bra et. al. [19]. We represent the structural characteristics of the user model by a class diagram (see Figure 4). The most detailed characterisation of the knowledge is its measuring for each user in each concept. However, this is not always the intent to have such a fine grained measuring. In such case, the knowledge is measured at higher level concepts.

Sometimes the structural view is not enough to model the tracking the user level of knowledge, his goals and preferences. Within a complex domain with many relationships, we propose to express the tracking of the user model by behavioural model. A state machine is used for expressing the states or levels of knowledge and transitions between them (e.g., when the preference or goal is changed).

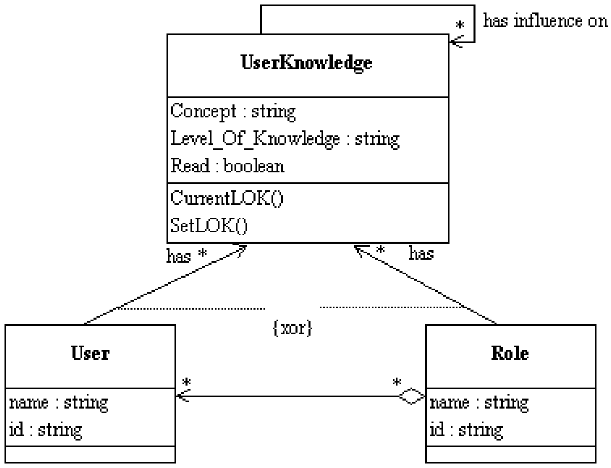

Fig. 4: An example of user model for e-lecture.

5 Presentation modelling

The model of presentation should reflect following characteristics: presentational form for information (font styles and colour, background colour, etc.); arrangement of the information chunks within the interface of the system; the way of the interface change according the user interaction with the system. Adaptive presentation modelling is based on rules for adaptation according to the user model or implicit rules, which do not depend on the user model.

In order to model a presentation we refined state machine developed within the navigational model by incorporating additional states (alternative or parallel), which represents different form of presentation for information. Rules, which express the adaptation according to the level of user knowledge are modelled by guards, conditions, and action of states and transitions. For example, considering the navigation model from Figure 3, the presentation model extends the *Domain Knowledge* state with two possible presentation states: *definition* and *detail description*, which appear according the level of user knowledge about this concept (see Figure 5).

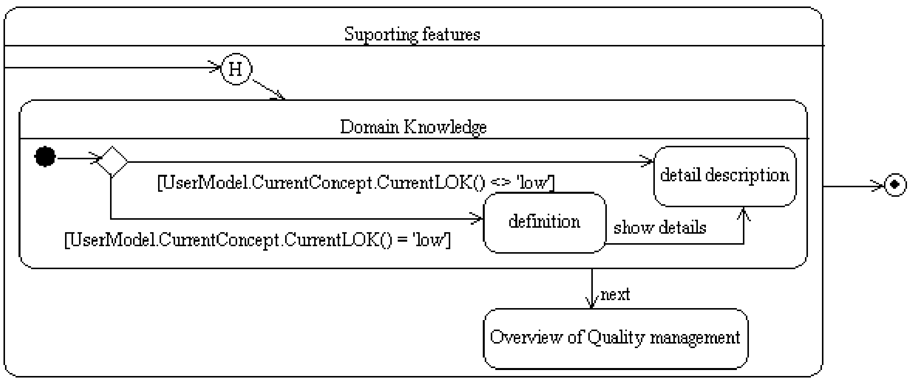

Fig. 5: A part of an adaptive navigation model for e-lecture.

The transition to the *detail description* state is labelled by the condition and corresponding action:

$$[UserModel.CurrentConcept.CurrentLOK() <>' low']/Display()$$

Next extension is setting the level of user knowledge after leaving a composite state:

$$nexttopic[UserModel.CurrentConcept.CurrentLOK() <>' low']/$$
$$Display(); UserModel.CurrentConcept.SetLOK()$$

The transition is labelled by three parameters: received event, guard and effect (action expression). First part is the next topic event. The guard condition related to the current level of the user knowledge is followed by the sequence of two actions that display related knowledge and set the new level of the user knowledge. The presentation model is interconnected with the user model by using the behavioural features `CurrentLOK()` and `SetLOK()` of the *UserKnowledge* class.

Special characteristic of a hypermedia system is that different media types are involved in. The concurrently presented media types have to be synchronised when they present relevant information. In our approach we model the channels [8] in presentation by composite states. The concurrent synchronised media channels are modelled by concurrent state and for synchronising these channels the synchronisation symbol is used (synchronisation or fork and join pseudo-metaclasses). All mentioned modelling elements are included in the state machine package of the UML.

Static presentational characteristics of classes or objects can be modelled by adding presentational structural features. Another possibility is to create new stereotype of class for presentation templates. These templates can be then mapped to particular states. We proposed and realised such solution in a system for creation of structured documents and annotations of their parts on the web [9].

6 Conclusions

We described the possible usage of UML for modelling of hypermedia application. We believe that state diagrams are better suited especially for personalised or adaptive hypermedia application modelling as techniques for structural modelling. The transition between states is semantically closer to an interaction based information change at the user interface. Structural techniques for navigation modelling use association or its stereotypes. This type of relationship is rather for modelling of static aspects of a system. Moreover, adaptation brings even more dynamics to the navigation.

In spite of this, static aspects of navigation modelling are also useful. As we mentioned, state diagrams are used only in HMBS/M and OOHDM but at the instance level. We allow to map also classes to states and relationships to transitions or state's decomposition. In addition, OOHDM and a lot of guidelines based on it recommend to use only structural techniques where only one "default browsing semantics" can be derived. However, this is not enough for interactive adaptive hypermedia application modelling where several browsing possibilities have to be considered for different users. Direct refinement of navigation model based on a state diagram can serve as the presentation model (different alternatives for content).

Further presented modelling techniques are used also in various existing methods for hypermedia development. Navigation modelling approach proposed in this paper requires different mapping rules between these techniques and the navigation model. Figure 6 depicts high level dependencies between models. These dependencies can be considered as a base for mentioned mappings. We are now working on precise definition of transformation of mentioned models from analysis to implementation.

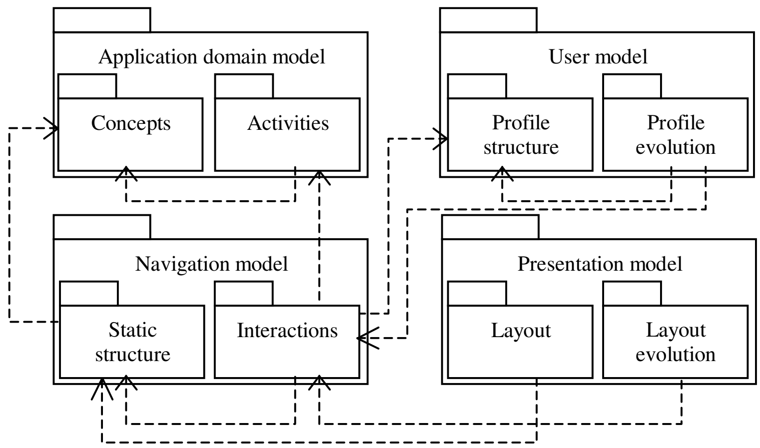

Fig. 6: High-level dependencies between hypermedia models.

This work was partially supported by Slovak Science Grant Agency, grant No. G1/7611/20.

Bibliography

1. L. Baresi, F. Garzotto, and P. Paolini. Extending UML for modeling web applications. In *Proc. of 34th Anual Hawaii International Conference on System Sciences (HICSS'34)*, Maui, Hawai, Jan. 2001. IEEE Press.
2. S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (WebML): a modeling language for designing web sites. *Computer Networks and ISDN Systems*, 33(1–6):137–157, June 2000.
3. J. Conallen. *Building Web Applications with UML*. Addison Wesley, 2000.
4. M. R. de Carvalho, M. C. F. de Oliveira, and P. C. Masiero. HMBS/M - an object oriented method for hypermedia design. In *Proc. of V Brazilian Symposium on Multimedia and Hypermedia Systems - SBMIDIA'99*, pages 43–62, Goiânia, June 1999.
5. P. Dolog. Modelling in hypermedia development, Aug. 2001. Technical Report (A Part of PhD Examination). Department of Computer Science and Engineering, Slovak University of Technology.
6. P. Dolog and M. Bieliková. Modelling browsing semantics in hypertexts using UML. In J. Zendulka, editor, *Proc. of ISM'2001 - Information Systems Modelling*, pages 181–188, Hradec nad Moravicí, Czech Republic, May 2001. MARQ, Ostrava.
7. P. Dolog and M. Bieliková. A statechart-based method for modelling adaptive navigation in hypermedia applications. In *Proc. of 4th Conference on Electrical Engineering and Information Technology for PhD Students*, Bratislava, Slovak Republic, Nov. 2001. Slovak University of Technology, Bratislava.
8. R. Červenka. Modelling of hypermedia applications. In B. Rovan, editor, *Proc. of SOFSEM'98 — 25th Conference on Current Trends in Theory and Practise of Informatics*, pages 294–303, Czech Republic, Nov. 1998. Springer LNCS 1251.
9. P. Fašung. Annotating structured documents on the web, Dec. 2001. Diploma Project supervised by Peter Dolog. Dept. of Computer Sci. and Eng., Slovak University of Technology.

10. F. Frasincar, G. J. Houben, and R. Vdovjak. A RMM-based methodology for hypermedia presentation design. In A. Caplinskas and J. Eder, editors, *Proc. of ADBIS 2001 - Advances in Databases and Information Systems*, pages 323–337, Vilnius, Lithuania, Sept. 2001. Springer, LNCS 2151.
11. P. Fraternali and P. Paoliny. Model-driven development of web applications: The autoweb system. *ACM Transactions on Information Systems*, 28(4):323–382, Oct. 2000.
12. R. Furuta and P. D. Stotts. A formally-defined hypertextual basis for integrating task and information, 1994. Tech. Report TAMU-HRL 94-007.
13. F. Garzotto and P. Paolini. HDM — a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1–26, Jan. 1993.
14. N. Guell, D. Schwabe, and P. Vilain. Modelling interaction and navigation in web applications. In *Proc. of the WWW and Conceptual Modeling Workshop*, Salt Lake City, 2000. Springer LNCS.
15. R. Hennicker and N. Koch. A UML-based methodology for hypermedia design. In A. Evans, S. Stuart, and B. Selic, editors, *Proc. of UML 2000 Conference*, York, England, Oct. 2000. Springer LNCS 1939.
16. IEEE. Guide to software engineering body of knowledge v. 0.95, 2001.
17. T. Isakowitz, A. Kamis, and M. Koufaris. Extending the capabilities of RMM: Russians dolls and hypertext. In *Proceedings of the 30th Annual Hawaii International Conference on System Sciences*, Jan. 1997.
18. D. Schwabe and G. Rossi. An object-oriented approach to web-based application design. *Theory and Practise of Object Systems (TAPOS), Special Issue on the Internet*, 4(4):207–225, Oct. 1998.
19. H. Wu, G. J. Houben, and P. D. Bra. User modeling in adaptive hypermedia applications. In *Proc. of the Interdisciplinaire Conferentie Informatiewetenschap*, pages 10–21, 1999.