Development Process and Evaluation Methods for Adaptive Hypermedia

Martin Balík and Ivan Jelínek
Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University
Karlovo náměstí 13, 121 35 Prague, Czech Republic

Keywords: Adaptive Hypermedia, Personalization, Development Process, Software Framework, Evaluation.

Abstract: Adaptive Hypermedia address the fact that each individual user has different preferences and expectations. Hypermedia need adaptive features to provide an improved user experience. This requirement results in an increased complexity of the development process and evaluation methodology. In this article, we first discuss development methodologies used for hypermedia development in general and especially for user-adaptive hypermedia development. Second, we discuss evaluation methodologies that constitute a very important part of the development process. Finally, we propose a customized development process supported by ASF, a special framework designed to build Adaptive Hypermedia Systems.

1 INTRODUCTION

Software development is a complex process, where modeling and specification on various levels have become a necessity and a standard approach. Web-based hypermedia systems require a special attention that has led to evolution of a new line of research – Web Engineering (Deshpande et al., 2002). A number of development methodologies have been created to offer new techniques, models and notations. Additional challenges came with a new category of intelligent, user-adaptive applications.

User-adaptive systems monitor users' behavior and keep track of each individual user's characteristics, preferences, knowledge, aims, etc. Some of the systems focus on providing the user with relevant items based on the browsing history. Other systems focus mainly on improving the human-computer interactions. The collection of personal data used in the adaptation process is associated with a specific user. It is called the User Model. While modeling the adaptive system, it is necessary to separate the non-adaptive and user-specific aspects of the application.

In our work, we focus on Adaptive Hypermedia Systems (AHS). Typical adaptation techniques used in AHS are categorized as *content adaptation*, *adaptive presentation*, and *adaptive navigation* (Knutov et al., 2009). The categories overlap, as some of the techniques do not change information or the possible navigation, but only offer suggestions to the user by changing the presentation. The design of adaptation techniques needs to be considered within the development process.

User-adaptive systems bring additional complexity into the development process and lay higher demands on system evaluation. This needs to be considered through all development phases. In order to guarantee the required behavior, we have to ensure that the system works correctly during and after adaptations (Zhang and Cheng, 2006).

Evaluation of adaptive systems is an important part of their development process and should not be underestimated. Currently, there is not much consistency in the evaluation of AHS (Mulwa et al., 2011). It is important to use an appropriate method for evaluation (Gena and Weibelzahl, 2007). Evaluation should ensure savings in terms of time and cost, completeness of system functionality, minimizing required repair efforts, and improving user satisfaction (Nielsen, 1993). AHSs are interactive, hypermedia-based systems. Usually, similar methods as in human-computer interaction (HCI) field are used. However, user-adaptive systems introduce new challenges.

The remainder of this paper is structured as follows. In Section 2, a current state of the art of development and evaluation methodologies is being reviewed. In Section 3, AHS development process is proposed and associated with the use of Adaptive System Framework. Finally, Section 4 concludes the paper by summarizing results of the research and indicates the directions of the future work.

Balík M. and Jelínek I..
Development Process and Evaluation Methods for Adaptive Hypermedia.
DOI: 10.5220/0004858401070114
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 107-114
ISBN: 978-989-758-024-6

2 RELATED WORK

In this section, we will review existing approaches used in AHS development. First, we will focus on the development methodologies mainly focused on design and system architecture. Second, we will review evaluation methodologies and problems related specifically with user-adaptive system evaluation.

2.1 AHS Development Methodologies

Similar to development of other software products, adaptive-system development needs to be based on standardized methods. For the design of hypermedia applications, several methods have been developed. In the early period of hypermedia systems, hypermedia-specific design methodologies were proposed, for example, Hypermedia Design Method (HDM) (Garzotto et al., 1993), Relationship Management Methodology (RMM) (Isakowitz et al., 1995), Enhanced Object-Relationship Model (EORM) (Lange, 1994) and Web Site Design Method (WSDM) (De Troyer and Leune, 1998). An Overview of additional and more recent development methodologies for software and Web engineering can be found in (Aragón et al., 2013; Thakare, 2012). However, the methodologies developed for hypermedia systems in general do not take into account the adaptivity and user modeling. Therefore, an extended adaptation-aware methodology is needed to improve the AHS development process.

Fig. 1 shows the typical phases of a software-development process. To abstract complex problems of the system design, models are used. The models help to create and validate the software architecture.

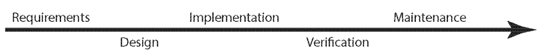

Figure 1: Typical phases of a software devel. process.

Model-Driven Architecture (MDA) (Miller and Mukerji, 2003) was proposed by the Object Management Group (OMG) in 2001. This architecture defines four model levels. *Computation-Independent Model (CIM)* describes behavior of the system in a language appropriate for users and business analysts. This level includes models of requirements and business models. *Platform-Independent Model (PIM)* is still independent of a specific computer technology, yet unlike the CIM it includes information essential for solving the assignment using information technologies. The PIM is usually created by computer analyst. The benefit of this level is the reusability for various implementations and platform independency. *Platform-Specific Model (PSM)* combines the PIM with a particular technology-dependent solution. This model can include objects tightly related to a specific programming language environment, e.g., constructors, attribute accessors, or references to classes included in the development platform packages. The model is an abstraction of source code structure and is used as a base for implementation. *Code* is the highest level of MDA and includes the implementation of the system.

Adaptive systems usually access large information base of domain objects, and their behavior is based on information stored in the user model. Such systems are quite complex and therefore, development methodology oriented on adaptive hypermedia is needed.

Object-oriented approach in designing adaptive hypermedia systems seems to be the most appropriate. Object oriented design is best suited for systems undergoing complex transitions over time (Papasalouros and Retalis, 2002). For object-oriented software systems modeling, we have a standard, widely-adopted, formally defined language – UML (Booch et al., 1999). To be able to express a variety of system models, UML provides extension mechanisms in definition of the model elements, description of the notation and expressing semantic of models. These extensions are stereotypes, tagged values and constraints. UML stereotypes are the most important extension mechanism.

There are some projects that utilize UML modeling in the area of adaptive systems. The Munich Reference Model (Koch and Wirsing, 2001) is an extension of the Dexter model. It was proposed in the same period as the well-known Adaptive Hypermedia Application Model (AHAM) (De Bra et al., 1999) and in a similar way adds a user model and an adaptation model. The main difference between The Munich Reference Model and AHAM is that AHAM specifies an adaptation rule language, while The Munich Reference Model uses object-oriented specification. It is described with the Unified Modeling Language (UML) which provides the notation and the object-oriented modeling techniques.

Object-Oriented Hypermedia Design Method (OOHDM) (Rossi and Schwabe, 2008) is based on both HDM and the object-oriented paradigm. It allows the designer to specify a Web application by using several specialized meta-models. OOHDM proposed dividing hypermedia design into three models – a conceptual model, a navigational model and an abstract interface model. When used to design a user-adaptive application, most of the personalization aspects are captured in the conceptual model.

As an example, we can mention a class model of the user and user group models (Barna et al., 2004).

Another method to specify design of complex Web sites is WebML (Ceri et al., 2000). For the phase of conceptual modeling, WebML does not define its own notation and proposes the use of standard modeling techniques based on UML. In the next phase, the hypertext model is defined. This model defines the Web site by means of two sub-models – composition model and navigation model. Development of the presentation model defining the appearance of the Web site is the next step. Part of the data model is the personalization sub-schema. The content management model specifies how is the information updated dynamically based on user's actions. Finally, the presentation model specifies how the system has to be adapted to each user's role (Aragón et al., 2013).

For the purpose of interoperability, storage models can be represented by a domain ontology. Therefore, there is a need to represent ontology-based models in a standardized way. Researchers already identified this issue and proposed UML profile for OWL and feasible mappings, which support the transformation between OWL ontologies and UML models and vice versa (Brockmans et al., 2006). This is achieved by the UML stereotypes. Table 1 provides the mappings for the most important constructs.

Table 1: UML and OWL mappings (Brockmans, 2006).

UML Feature	OWL Feature	Comment
class, type	class	
instance	individual	
ownedAttribute, binary association	property, inverseOf	
subclass, generalization,	subclass, subproperty,	
N-ary association, association class	class, property	Requires decomposition
enumeration	oneOf	
disjoint, cover	disjointWith, unionOf	
multiplicity	minCardinality, maxCardinality, FunctionalProperty, InverseFunctionalProperty	OWL cardinality restrictions declared only for range
package	ontology	

Special attention should be also devoted to the development of the content of the adaptive systems. As it was observed many times – authoring of adaptive systems is a difficult task (Cristea, 2003). The adaptive-system development process can be divided into four phases: Conceptual Phase, Presentation Phase, Navigation Phase and Learning Phase (Medina et al., 2003).

During the conceptual phase, the author creates basic page elements, in the presentation phase the structure of page elements is defined, in the navigation phase the navigational map is created and in the learning phase, adaptive behavior is defined.

2.2 AHS Evaluation Methodologies

Recent research has identified the importance of user-adaptive systems evaluation. Reviews on the topic have been published by several researchers (Gena, 2005; Velsen et al., 2008; Mulwa et al., 2011; Albert and Steiner, 2011). Due to the complexity of adaptive systems, the evaluation is difficult. The main challenge lies in evaluating particularly the adaptive behavior. Evaluation of adaptive systems is a very important part of the development process. Moreover, it is necessary, that correct methods and evaluation metrics are used.

Usability is evaluated by the quality of interaction between a system and a user. The unit of measurement is the user's behavior (satisfaction, comfort) in a specific context of use (Federici, 2010). Design of adaptive hypermedia systems might violate standard usability principles such as user control and consistency. Evaluation approaches in HCI assume that the interactive system's state and behavior are only affected by direct and explicit action of the user (Paramythis et al., 2010). This, however, is not true in user-adaptive systems.

Personalization and user-modeling techniques aim to improve the quality of user experience within the system. However, at the same time these techniques make the systems more complex. By comparing the adaptive and non-adaptive versions, we should determine the added benefits of the adaptive behavior.

General (non-adaptive) interactive systems acquire from user the data strictly related to the performed task. Adaptive systems, however, require much more information. This information might not be required for the current task and can be in the current context completely unrelated. This is caused by continuos observation of the user by the system. Adaptive systems can monitor visited pages, keystrokes or mouse movement. Users can be even asked superfluous information directly. Within the evaluation process, it is challenging to identify the purpose and correctness of such a meta-information.

Important difference between evaluation of adaptive and non-adaptive systems is that evaluation of adaptive systems cannot consider the system as a whole. At least two layers have to be evaluated separately (Gena, 2005).

In the next paragraphs, we will summarize the most important methods used to evaluate adaptive hypermedia systems.

Comparative Evaluation

It is possible to assess the improvements gained by adaptivity by comparing the adaptive system with a non-adaptive variant of the system (Höök, 2000). However, it is not easy to make such comparison. It would be necessary, to decompose the adaptive application into adaptive and non-adaptive components. Usually adaptive features are an integral part of the system, and the non-adaptive version could lead to unsystematic and not optimal results. Additionally, it might not be clear why the adaptive version is better.

In case of adaptive learning, a typical application area of adaptation, it is possible to compare the system with a different learning technology or with traditional learning methods. However, the evaluation of adaptation effects can interfere with look and feel or a novelty effect (Albert and Steiner, 2011).

Empirical Evaluation

Empirical evaluation, also known as the controlled experiment, appraises theories by observations in experiments. This approach can help to discover failures in interactive systems, that would remain uncovered otherwise. For software engineering, formal verification and correctness are important methods. However, empirical evaluation is an important complement that could contribute for improvement significantly. Empirical evaluation has not been applied for the user modeling techniques very often (Weibelzahl and Weber, 2003). However, in recent studies, the importance of this approach is pointed out (Paramythis et al., 2010). This method of evaluation is derived from empirical science and cognitive and experimental psychology (Gena, 2005). In the area of adaptive systems, the method is usually used for the evaluation of interface adaptations.

Layered Evaluation

For evaluation of adaptive hypermedia systems, usually approaches considering the system “as a whole” and focusing of an “end value” are used. Examples of the focused values are user’s performance or users’s satisfaction. The problem of this approach is, that evaluating system as a whole requires building the whole system before evaluation. This way, the evaluation is not able to guide authors in the development process. Another problem is, that the reasons behind unsatisfactory adaptive behavior are not evident.

A solution to the mentioned problems was proposed by Brusilovsky in (Brusilovsky and Sampson, 2004) as a model-based evaluation approach called *layered evaluation*. In the exemplary case, two layers were defined – user modeling layer and adaptation decision making layer. User modeling (UM) is the process, where information about user is acquired by monitoring user-computer interaction. Adaptation decision making is a phase, where specific adaptations are selected, based on the results of the UM phase. Both processes are closely interconnected. However, when evaluating the system as a whole, it is not evident, which of the phases has been unsuccessful. This is solved by decomposing evaluation into layers and evaluating both phases separately. This has also the benefit, that results of UM process evaluation can be reused for different decision making modules.

Layered evaluation has gained a high level of attention in the adaptive hypermedia research community. That reaffirms the claim that the evaluation of adaptive systems implicates some inherent difficulties (Mulwa et al., 2011). The original idea is often used by authors to justify experimental designs of their evaluation studies.

Process-oriented Evaluation

Evaluation should be considered as an inherent part of the development cycle. Continuous evaluation should range from very early phases of the project till the end. Evaluation should start with requirements analysis and continue at the prototype level. Evaluation of initial implementations is referred as *formative evaluation*. Identifying early issues can greatly reduce development costs. The quality of the overall system is evaluated in the final phase of the development cycle and is referred a *summative evaluation*. The focus of current evaluations of adaptive systems is mostly targeted on the summative evaluation. To ensure that user’s needs are sufficiently reflected, formative evaluation must be more intensively used.

User-centered Evaluation

For adaptive systems, especially user-centered evaluation approaches are recommended (Velsen et al., 2008).

Following are the typical user-centered evaluation methods:

- **Questionnaires**
  Questionnaires collect data from users by answering a fixed set of questions. They can be used to collect global impressions or to identify problems. Advantage is, that large number of participants can be accommodated (compared to interviews).
- **Interviews**
  In interviews, participants are asked questions by an interviewer. Interviews can identify individual and situational factors and help explain, why a system will or will not be adopted.

- **Data Log Analysis**
  The log analysis can focus on user behavior or the user performance. It is strongly advised to use this method with a qualitative user-centred evaluation.
- **Focus Groups and Group Discussions**
  Groups of participants discuss a fixed set of topics, and the discussion is led by a moderator. This method is suitable for gathering a large amount of qualitative data in a short time.
- **Think-aloud Protocols**
  Participants are asked to say their thoughts out loud while using the system.
- **Expert Reviews**
  System is reviewed by an expert, who gives his opinion.

3 AHS DEVELOPMENT PROCESS

The development methodologies mentioned in section 2.1 were developed for non-adaptive hypermedia systems and therefore, the methodologies do not provide sufficient support for the adaptation process. By adding adaptive features, the design complexity increases. Without adequate development support, the application can become unmaintainable, or the behavior of the application can become inconsistent.

As an example of deficiency, the OOHDM methodology allows user-role-based personalization as part of the conceptual model. However, there is no clear separation of the user-adaptive behavior. Although the WebML defines an explicit personalization model for users and user-groups, it is missing means for expressing and separating various adaptation methods. Other legacy development methodologies do not consider personalization at all.

In a development methodology, two important components can be identified. One of them is the language, which can be used by a designer to model the different aspects of the system. The other component is the development process, which acts as the dynamic, behavioral part. The development process determines what activities should be carried out to develop the system, in what order and how. To specify the development process for user-adaptive hypermedia systems, we follow the model-driven architecture (MDA).

Fig. 2 depicts the MDA adopted to user-adaptive hypermedia systems engineering. The principles are visualized as a stereotyped UML activity diagram based on the diagram presented in (Koch et al., 2006). The process starts with the Computation-Independent Model (CIM) that defines requirements models and user characteristics model. Platform-Independent Model (PIM) is divided into two segments. User Independent Model (UIM) describes the system without its adaptation features and is equivalent to the standard web engineering design methodology. Three models, based on the OOHDM, are created – conceptual model, navigational model, and abstract interface model. The other segment consists of the User Specific Model (USM). USM consists of three sub-models, that are patterned on adaptation method categories (Knutov et al., 2009) – content adaptation model, adaptive navigation model, and adaptive presentation model.

The user-specific PIM sub-models are closely related with our theoretical basis of adaptive hypermedia architecture – the Generic Ontology-based Model for Adaptive Web Environments (GOMAWE). The adaptation function, defined as a transformation between default and adapted hypermedia elements, is the basis for content adaptation. Transformations are defined by Inference Rules. Adaptive navigation defines transformations within the navigational model, and results into the Link-Adaptation Algorithms in subsequent modeling phases. Adaptive presentation is modeled as transformations within the Adaptive Hypermedia Document Template. For formal definitions of GOMAWE, see (Balík and Jelínek, 2013b).

After the models for both the user-independent and user-specific segments are separately defined, they can be transformed and merged together to form the "big picture" of the system. The next step is transforming the PIM into the Platform-Specific model (PSM). As an example, we show Java and .NET model, but there are many other possible platforms. From the PSM, a program code can be possibly generated.

While the PIM depends usually in large extent on UML and UML profiles that provide a standard abstract model notation, the PSM, on the other hand, should refer to software framework packages used to simplify the development on a specific platform. In our previous work, we have proposed a software framework intended to support the development of user-adaptive hypermedia systems. The Adaptive System Framework (ASF) (Balík and Jelínek, 2013a) defines a fundamental adaptive hypermedia system architecture and implements the most common adaptive system components.

One of the important ASF components is the user-specific data storage. The centralized user model management is beneficial for the application development. Using the adaptation manager, the user profile and user model properties can be accessed from any component of the application. Another part of the data-storage layer is the rule repository. A rule-repository manager provides an interface for accessing and evaluating the inference rules. This interface can be utilized in the adaptation algorithms, e.g., the content adaptation algorithm can use conditional rules to find an alternative content for a specific user.

The design of the application core based on the ASF framework consists of the following important steps:

1. Definition of the domain objects and their relations
2. Definition of the user profile and user model attributes
3. Design of the adaptive algorithms for the desired behavior
4. Configuration of data sources
5. Binding the data results either to the application logic or directly to the adaptive UI components

All the steps are supported by the ASF framework. Based on UML model, the developer implements domain objects by using support classes of the framework. User data storage needs only data model specification (preferably as an ontology). Adaptive algorithms can be reused or extended. And finally, user interface components can be used to support the presentation.

The implemented user-adaptive application needs to be evaluated, and evaluation should be an integral part of the development process. Various methods mentioned in Section 2.2 can be used.

Based on the evaluation methodology proposed in (Lampropoulou et al., 2010), we use a three-phase evaluation as part of the development process. The first phase is a short empirical study, in the second phase a qualitative and quantitative measurement is performed, and finally, the third phase evaluates subjective comments of test session participants. For the purpose of AHS evaluation, we extend the second and third phases by the comparison with a non-adaptive system users control group.

Typical adaptive system evaluation is based on comparison between adaptive and non-adaptive version of the application. ASF framework is well

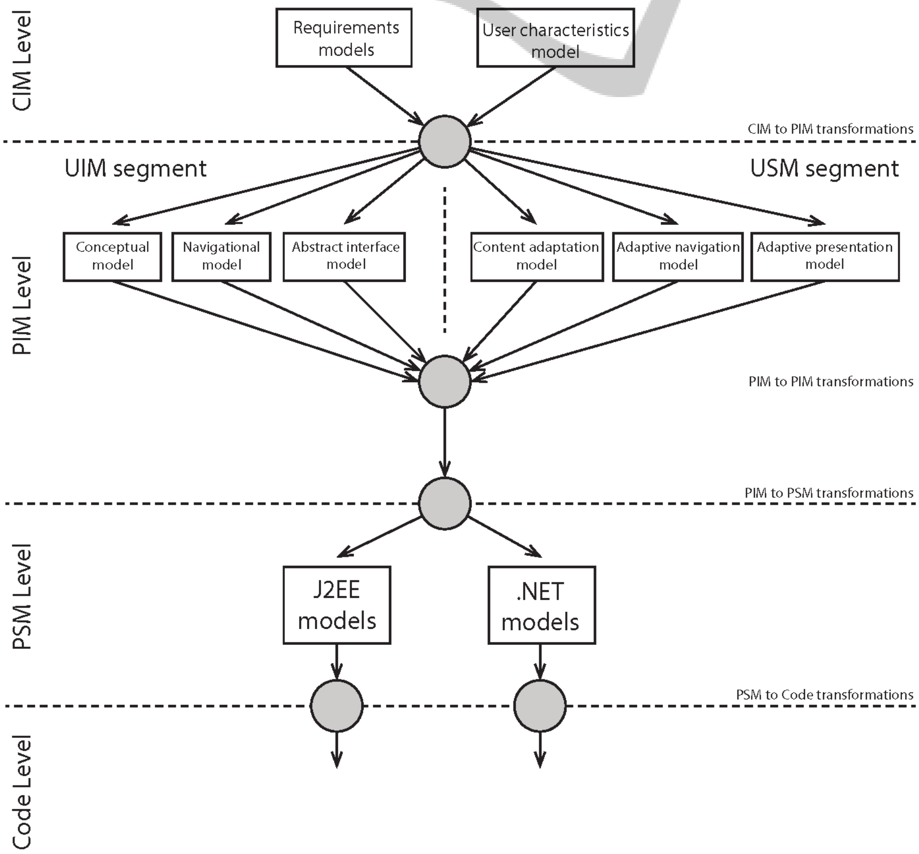

Figure 2: MDA structure for user-adaptive hypermedia systems engineering.

designed for such a comparative evaluation. Each adaptation-type algorithm strategy supports the non-adaptive algorithm version. This feature can be used as an additional user-accessible preference setting, or it can be administered for special purposes, e.g., to support the adaptation evaluation session.

To be reasonable, the evaluation needs to be performed with a representative group of users. For such purposes, adaptive educational applications, where a large amount of students can participate, is highly appropriate. Many of the typical aspects of adaptive applications can be simulated and evaluated by students. The tutorials can include theoretical tests, practical assignments, or test questions used to review the knowledge of students.

In our adaptive e-learning prototype, we focus mainly on the user-centered evaluation. In the fist evaluation phase, the students were asked about their preferences regarding the online curriculum. The questions included preference of used adaptation techniques. They were also asked if their results should be available to the tutor with all details, in a form of whole class statistics, or completely hidden. In the second evaluation phase, we used a data log analysis to observe the behavior of users, progress in knowledge and selected preferences. The session with multiple students is suitable to measure the system performance, identify possible bottlenecks and compare the adaptive system with the non-adaptive alternative. The evaluation sessions are usually combined with questionnaires, where students answer questions related to the application content, and they can provide a feedback about their satisfaction or issues they encounter while using the system. This is the last of the three evaluation phases. Afterwards, all the collected data are analyzed, and the results provide a feedback for system customization and improvements.

4 CONCLUSION

Design, modeling and evaluation are fundamental steps in the development process of software products. Web technologies and requirements of personalization add more complexity into the process, and specialized methodologies are needed. In this paper, we have given an overview of existing methodologies and their use in the context of user-adaptive systems. Further, we have proposed a special methodology for adaptive hypermedia, based on MDA and OOHDM. The development methodology was extended to include the aspects of user-adaptive systems. The AHS-specific methodology is important for improving the development effectiveness and quality of the resulting product.

In our future work, we aim to use the methodology in additional prototypes' development based on ASF. We will apply the framework in different application types, and we will focus in more detail on adaptive systems' recommendation adaptation features. Further, we want to integrate the learning curriculum application with other systems and assessments used in the courses, and we want to utilize the ontology-based data maintained by the adaptive systems to exchange the user models of students.

ACKNOWLEDGEMENTS

The results of our research form a part of the scientific work of a special research group WEBING.[1]

REFERENCES

Albert, D. and Steiner, C. M. (2011). Reflections on the Evaluation of Adaptive Learning Technologies. In *Proceedings of the IEEE International Conference on Technology for Education (T4E)*, pages 295–296.

Aragón, G., Escalona, M.-J., Lang, M., and Hilera, J. R. (2013). An Analysis of Model-Driven Web Engineering Methodologies. *Int. Journal of Innovative Computing, Information and Control*, 9(1):413–436.

Balík, M. and Jelínek, I. (2013a). Adaptive System Framework: A Way to a Simple Development of Adaptive Hypermedia Systems. In *The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE2013)*, pages 20–25, Valencia, Spain. IARIA.

Balík, M. and Jelínek, I. (2013b). Generic Ontology-Based Model for Adaptive Web Environments: A Revised Formal Description Explained within the Context of its Implementation. In *Proceedings of the 13th IEEE International Conference on Computer and Information Technology (CIT2013)*, Sydney, Australia.

Barna, P., Houben, G.-j., and Frasincar, F. (2004). Specification of Adaptive Behavior Using a General-purpose Design Methodology for Dynamic Web Applications. In *Proceedings of AdaptiveHypermedia and Adaptive Web-Based Systems (AH 2004)*, Eindhoven, The Netherlands.

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*, volume 30 of *Addison-Wesley object technology series*. Addison-Wesley.

Brockmans, S., Colomb, R. M., Haase, P., Kendall, E. F., Wallace, E. K., Welty, C., and Xie, G. T. (2006). A model driven approach for building OWL DL and

[1]Webing research group – http://webing.felk.cvut.cz

OWL full ontologies. In *ISWC'06 Proceedings of the 5th international conference on The Semantic Web*, pages 187–200. Springer-Verlag.

Brusilovsky, P. and Sampson, D. (2004). Layered evaluation of adaptive learning systems. *Int. J. Continuing Engineering Education and Life-Long Learning*, 14(4-5):402–421.

Ceri, S., Fraternali, P., Bongio, A., and Milano, P. (2000). Web Modeling Language (WebML): a modeling language for designing Web sites. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33(1-6):137–157.

Cristea, A. (2003). Automatic authoring in the LAOS AHS authoring model. In *Hypertext 2003, Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems*, Nottingham, UK.

De Bra, P., Houben, G.-J., and Wu, H. (1999). AHAM : A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 147–156. ACM.

De Troyer, O. and Leune, C. (1998). WSDM: a user centered design method for Web sites. In *Proceedings of the Seventh International WWW Conference*, volume 30, pages 85–94, Brisbane, Australia. Elsevier Science Publishers B. V.

Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., and White, B. (2002). Web engineering. *Journal of Web Engineering*, 1(1):3–17.

Federici, S. (2010). Usability evaluation : models, methods, and applications. In *International Encyclopedia of Rehabilitation*.

Garzotto, F., Schwabe, D., and Paolini, P. (1993). HDM A Model-Based Approach to Hypertext Application Design. *ACM Trans. on Information Systems*, 11(1):1–26.

Gena, C. (2005). Methods and techniques for the evaluation of user-adaptive systems. *The Knowledge Engineering Review*, 20(01):1.

Gena, C. and Weibelzahl, S. (2007). Usability Engineering for the Adaptive Web. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The adaptive web*, pages 720–762. Springer-Verlag, Berlin, Heidelberg.

Höök, K. (2000). Steps to take before intelligent user interfaces become real. *Interacting with Computers*, 12(4):409–426.

Isakowitz, T., Stohr, E. A., and Balasubramanian, P. (1995). RMM: a methodology for structured hypermedia design. *Communications of the ACM*, 38(8):34–44.

Knutov, E., De Bra, P., and Pechenizkiy, M. (2009). AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, 15(1):5–38.

Koch, N. and Wirsing, M. (2001). Software Engineering for Adaptive Hypermedia Applications? In *8th International Conference on User Modeling*, pages 1–6, Sonthofen, Germany.

Koch, N., Zhang, G., and Escalona, M. J. (2006). Model Transformations from Requirements to Web System Design. In *Proc. Sixth Int'l Conf. Web Eng.*, pages 281–288.

Lampropoulou, P. S., Lampropoulos, A. S., and Tsihrintzis, G. A. (2010). A Framework for Evaluation of Middleware Systems of Mobile Multimedia Services. In *The 2010 IEEE International Conference on Systems, Man, and Cybernetics (SMC2010)*, pages 1041–1045, Istanbul, Turkey.

Lange, D. (1994). An Object-Oriented Design Method for Hypermedia Information Systems. In *Proceedings of the 27th - Hawaii International Conference on System Sciences*, volume 6, pages 336–375, Hawaii. IEEE Computer Society Press.

Medina, N., Molina, F., and García, L. (2003). Personalized Guided Routes in an Adaptive Evolutionary Hypermedia System. In *Lecture Notes in Computer Science 2809*, pages 196–207.

Miller, J. and Mukerji, J. (2003). MDA Guide Version 1.0. [Accessed: August 24, 2013]. [Online]. Available: http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf.

Mulwa, C., Lawless, S., Sharp, M., and Wade, V. (2011). The evaluation of adaptive and personalised information retrieval systems: a review. *International Journal of Knowledge and Web Intelligence*, 2(2-3):138–156.

Nielsen, J. (1993). *Usability Engineering*. Academic Press, Boston, MA.

Papasalouros, A. and Retalis, S. (2002). Ob-AHEM: A UML-enabled model for Adaptive Educational Hypermedia Applications. *Interactive educational Multimedia*, 4(4):76–88.

Paramythis, A., Weibelzahl, S., and Masthoff, J. (2010). Layered evaluation of interactive adaptive systems: framework and formative methods. *User Modeling and User-Adapted Interaction*, 20(5):383–453.

Rossi, G. and Schwabe, D. (2008). Modeling and implementing web applications with OOHDM. In Rossi, G., Pastor, O., Schwabe, D., and Olsina, L., editors, *Web Engineering*, chapter 6, pages 109–155. Springer.

Thakare, B. (2012). Deriving Best Practices from Development Methodology Base (Part 2). *International Journal of Engineering Research & Technology*, 1(6):1–8.

Velsen, L. V., van der Geest, T., Klaasen, R., and Steehouder, M. (2008). User-centered evaluation of adaptive and adaptable systems: a literature review. *The Knowledge Engineering Review*, 23(3):261–281.

Weibelzahl, S. and Weber, G. (2003). Evaluating the inference mechanism of adaptive learning systems. In *User Modeling 2003*, pages 154–162. Springer-Verlag.

Zhang, J. and Cheng, B. H. C. (2006). Model-based development of dynamically adaptive software. In *Proceeding of the 28th international conference on Software engineering - ICSE '06*, number 1, pages 371–380, New York, New York, USA. ACM Press.