Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science and Engineering

Ontology-based Model for Personalized Web Design

by

Martin Balík

A thesis submitted to
the Faculty of Electrical Engineering, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

PhD programme: Electrical Engineering and Information Technology
Specialization: Computer Science and Engineering

November 2013

Thesis Supervisor:
Ivan Jelínek
Department of Computer Science and Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo nám. 13
121 35 Praha 2
Czech Republic

Abstract and contributions

The amount of information on the web is continually growing. Orientation within accessible information is becoming more and more difficult for the user. Therefore, there is a need to personalize the presented information and make it comply with the user's expectations. Adaptation is often discussed within the field of web technologies. Intensive research within the field of adaptive systems has been carried out in the last two decades. Several models have been proposed for the description of adaptive hypermedia architecture. However, there is still a lack of generality in the architecture, which makes collaboration and content reusability difficult, even impossible. Most of the contemporary approaches use ad-hoc solutions and there is a need of a general formal model to simplify development of adaptive systems and enable data interchange among them. In our work, we aim to propose such a model, its formal description and the methodology for developing systems based on this model as a contribution to the current state of research in the field of hypermedia systems.

The main contributions of the thesis are the following:

1. New formal adaptivity model utilizing semantic web technologies for user modeling in adaptive hypermedia systems.

2. Multidimensional user model architecture supporting effective reasoning and data exchange.

3. Design of data exchange format based on XML and RDF.

4. Methodology for effective adaptive systems development.

5. Framework support for adaptive systems implementation in Java programming language and its experimental verification.

Keywords:

adaptive hypermedia, personalization, user modeling, general model, formal description, Semantic Web, ontologies, interoperability, e-learning

Acknowledgements

First of all, I would like to express my gratitude to my thesis supervisor Ivan Jelínek. He has been a constant source of encouragement and insight during my research.

I would like to thank the Department of Computer Science and Engineering. Especially, I would like to thank to Filip Železný, Miroslav Šnorek, and prof. Pavel Tvrdík for taking care of my financial support, while they worked at the position of head of the department.

My work has been partially supported by grants: Grant of Czech Grant Agency for 2006-2008(Intelligent Web Technologies - 201/06/0648), grant from Czech Technical University (IGS) in years 2007 (CTU0706713), 2008 (CTU0811813) and 2009 (CTU0915213). I would like to thank to relevant agencies for their financial support that has contributed to the research results.

I would also like to thank Miroslav Bureš for reading my thesis and for his constructive comments.

Finally, my greatest thanks to my mother and to my girlfriend Jana whose support helped me in this work.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

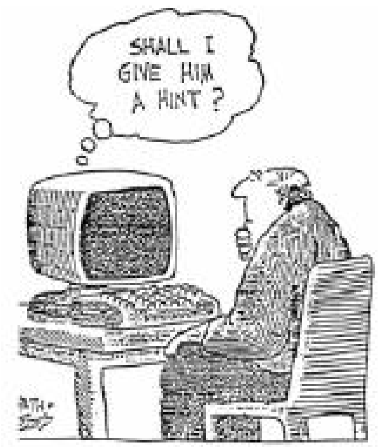


Figure 1.1: Intelligent Computer [139].

1.1 The need of user adaptive systems

Imagine a news web portal providing too much information. Alternatively, imagine an e-shop selling a large amount of goods. Every user is interested in different types of information and has variable preferences. Typical hypermedia systems display the same content for each user. Common content is unsuitable for all, because users differ substantially in many aspects. This problem prevents users from working effectively. There is a need for personalization – process, which selects the information important for the individual user. Every user prefers different browsing paths through web information. The solution is a system having the ability to adapt its behavior to the goals, tasks and interests, as well as other features of individual users and groups of users [27].

Humans constantly monitor the world around them. Computers, however, are built to do what they are told to do, nothing more, and nothing less. Generally, computers do not exhibit such modeling behavior as humans do [61]. Adaptive systems should remedy this by storing a user model containing each user's preferences. User adaptive systems perform incremental behavior analysis to model the user. Using the stored information the adaptation is performed. This process is performed during the interaction of the user with the system. This means that we can consider adaptive systems to be interactive.

Many researchers are working on the development of solutions for content and navigation adaptation of hypermedia spaces. Adaptive systems are mostly used in the fields of e-commerce [30] or e-learning [24]. Several models have been proposed for the description of adaptive hypermedia architecture. However, most of the current systems are developed using ad-hoc approaches and as a result are unable to cooperate and reuse their data.

1.2 Motivation

In this section, we will explain the motivation and aims of our work.

We have identified that there are no generally accepted standards on adaptive personalization of hypermedia systems. Adaptive hypermedia systems are usually developed ad-hoc, and custom architectures are designed whenever personalization is required. This results in heterogeneous user model representations and incompatible data formats. Terminology usually differs, and systems with similar functionality are developed over again.

Reference models developed in the beginning of adaptive hypermedia (AH) era were designed for small to medium-size adaptive applications and do not consider more recent technologies. Semantic and data-oriented approaches, social networks and mobile applications are new trends extending the possibilities in AH. At the same time, with the use of modern technologies, new challenges are emerging. To overcome the missing parts, not included in the conventional reference models, extension is needed. We will identify important features for current AH systems, related extensions of the adaptation process will be explained, and the changes will be reflected into the up-to-date reference model.

Another problem of contemporary adaptive systems is insufficiency of data interoperability and reusability. Individual systems suffer by the cold-start problem. Central user-modeling server can cause maintainability problems and integration with an existing application is impossible. It is feasible to solve interoperability issues by the utilization of Semantic Web technologies. The adaptation quality of individual applications can be improved by sharing users' preferences. Ontologies are just the appropriate data representation to achieve interoperability and therefore, ontologies need to be part of the AH reference model.

We have identified similar reusability problem within the development process of adaptive hypermedia systems (AHS). Web-based applications are complex and depend on wide range of technologies. To improve the development eficiency, there exists a support for many common development problems. Architects follow development methodologies, the implementation support is provided by software libraries and frameworks. Libraries can solve problems such as security, e.g. Spring-security, data persistence, e.g. Hibernate, or communication, e.g. Jersey. However, there is no generally applicable support for advanced hypermedia personalization. Such support is one of the aims of our work.

The heterogeneity of AH systems was a motivation to create a new reference model of AHS, extended and general enough to capture important abstractions found in the existing and future Adaptive Hypermedia Systems. Our research scope is focused on software engineering, development and architecture. The main contributions are the novel definition of adaptive software product components and the adaptation-aware development methodology enhancement. Our aim is to provide an extensible plaform for various adaptation methods and diverse data-modeling approaches. Introducing new methods of information processing or semantic reasoning is not the primary aim of this thesis, still the merits of logic and computational science are required for an evaluable implementation based on our proposals.

1.2.1 Research questions

To outline the necessary steps of our research, we have asked several questions that we are going to answer in our work.

- **Why is the generic ontology-based model necessary? Is it possible to devise the model from existing AHS?**
  We have outlined the motivation for our research in the previous section. To show the necessity of the generic ontology-based AHS model, we will discuss various features and requirements of existing AHS. The generic ontology-based model will be defined based on the requirements. To show feasibility to devise a generic ontology-based AHS model, several use-case studies will be used to show the research steps necessary to devise the model. Finally, the use-cases will show the compliance of applications with the model.

- **What are the elementary components of adaptive systems?**
  We answer this question by studying existing adaptive systems. We review existing methods and techniques used for personalization. We look at existing reference models, AHS implementations based on the models and their application in various fields, where hypermedia is used. To create a generic AH architecture, we had to analyze existing solutions, and extract core functionality to well defined components. Extensions of contemporary reference models will be introduced.

  The evolution of the novel reference model required to follow up more specific sub-questions:

  - **What information about a user needs to be stored by the application to personalize the presentation?**
    To be able to define a generic AH framework, it is necessary to categorize user data captured within the system. We need to design an extendable user model supporting effective storage of personal characteristics.
  - **What information needs to be communicated while several adaptive systems cooperate and share user data?**
    To support integration of diverse adaptive systems, we need to define data structures containing the user-specific information and expose a suitable application interface. This will be covered within the explanation of the integration layer of the proposed model.
  - **What is the role of the Semantic Web in the area of adaptive hypermedia?**
    Semantic Web aims at making the current, mostly unstructured, documents machine understandable. Semantics has a significant impact in many research areas, including AH. The Chapter 2 will cover the review and com-

parison of various attempts to utilize semantic technologies in the field of AH. The gained benefits were evaluated by use-case applications and the reasonability of utilizing Semantic Web technologies in the development of AH systems will be discussed in the context of our work.

- **What methodologies should adaptive hypermedia developers follow?**
  Software development methodologies do formalize the communication between members of development teams to make the process more efficient and repeatable. Adaptive systems are software systems that can follow conventional methodologies. However, personalization needs a special attention. We will discuss suitable methodology for incorporation of adaptive behavior.

  The reference model itself supports the development of new adaptive systems, as it defines the system components that developers should follow. To contribute substantially to support the AHS development, core implementation of the model, development methodology and exemplary use-cases are the additional steps.

  The following sub-questions point out the most important aspects of AHS development:

  - **What architecture is suitable and generally applicable for various user-adaptive systems?**
    To be able to successfully develop a well-performing and easy-to-maintain adaptive system, we have to build the system on well established foundations. In our research, we focused on determining suitable software components and integrate them into a tool serving as general-purpose support for adaptive system development.
  - **What best-practice solutions should we apply within the development of AHS?**
    Software engineering history has proven that there exist several "best practice" solutions that is appropriate to follow. We keep this in mind when designing the Adaptive System Framework.
  - **How can we evaluate the quality of user adaptive system?**
    Evaluation of adaptive systems is not simple and currently not much unified. We try to answer this question by identifying several evaluable features of adaptive systems.

1.3 Contributions of the thesis

The thesis aims to find suitable solution for the problems described in the previous section. The results were published on a number of conferences and recently in a peer-reviewed journal. The main contributions of the thesis are the following:

1. We have reviewed the state of art in the area of adaptive hypermedia research, identified the requirements and possible directions of research. The review was published in [A.1] and recently extended to comply with the latest research. We have identified the most important requirements of adaptive hypermedia systems, and we have proposed an extended modeling loop in [A.2].

2. A new formal model of adaptive system architecture was proposed. The model is utilizing semantic web technologies for user modeling in adaptive hypermedia systems. The architecture of the model was published in [A.5] and the proposal was followed by a first experimental prototype implementation [A.7]. The contribution of ontologies in user modeling was presented in [A.4] and possible use case in area of adaptive learning was proposed in [A.3]. In 2013, the description of the revised formal specification of the model was accepted and will be published in December [A.11].

3. We aim to achieve an effective general solution for any system that needs to adapt to users. Multidimensional user model architecture supporting efficient reasoning and data exchange was designed for this purpose and presented in [A.5].

4. Adaptive systems need to exchange user data to avoid user diversity and cold start problem. Design of data-exchange format based on XML [A.9] and RDF is part of the solution for interoperability issues [A.6]. In 2013, a RESTful API for RDF-based User Model data was designed, and we plan to publish the results of our experiments in the near future, for details see Section 5.4.2.

5. There is a need for a complete and easy-to-follow methodology of the adaptive system development. Finally, every adaptive system needs to be verified and evaluated. Methodology for effective adaptive systems development and verification was proposed [A.9].

6. Following the formal model, a framework has been designed, and the framework is being implemented as a Java library with several layers. Core layer is

standalone and provides abstract concepts to higher layers. Selected adaptation algorithms are part of the framework library, and we have already performed and evaluated experiments focused on adaptation algorithms in [A.8]. Implementation dependent parts are based on a Spring framework and Primefaces user interface components. In 2013, the detailed description of the framework was published in [A.10].

1.4 Terminology

To avoid confusion, will define here the most common terms used in the area of adaptive hypermedia systems. Some of the terms were adopted from [58] and extended to reflect our conception of the problem domain.

1.4.1 Systems

The following terms denote the categories of computer systems in the scope of our research.

Personalization. Personalization is the activity where a system is changed to conform better to the user. This is typically performed by explicit user actions at one particular location in the system, e.g., a preference screen.

Adaptive personalization. In Adaptive personalization the personalization is based on a user model that is maintained adaptively. This means that the user's actions are observed by the system and used to base the user model. Data in the user model is used to personalize the presented information.

User adaptive system. A user adaptive system is a system that performs adaptive personalization.

Adaptable hypermedia system. Adaptable hypermedia systems are hypermedia systems, which allow users to customize a specific behavior and features of the system. This type of adaptation requires explicit user action to change the behavior, e.g., on a preference screen.

Adaptive hypermedia system. Adaptive hypermedia systems are hypermedia systems, which reflect some features of the user in a user model and use this model by adapting various visible aspects of the system to the user [22]. A system is called adaptive if it changes its behavior automatically according to its context.

1.4.2 Processes

The following terms define the process performed in the system.

User modeling. User modeling is the area of research that focuses on adaptive personalization. The term is, however, also used occasionally for user profiling, or to mean only the activity of obtaining user models. In software engineering, the term is used for the design activity that creates a model of the future users of the system. We will use it as a synonym of adaptive personalization.

User model. A model of the relevant characteristics of a user that is or can be used to personalize the behavior or presentation of a system.

User profiling. User profiling entails the use of a user profile to personalize a system. A user profile is provided manually beforehand either by the user himself, or by a third person. As such user profiling is not adaptive.

User profile. A set of parameters that are predefined or changed by the user himself. These are usually accessible as a user preference screen.

1.4.3 Components

The following terms define the system components and data structures.

Adaptation component. An adaptation component is the component in a user adaptive system that takes care of maintaining a user model and using this model to answer the questions about the user necessary to perform personalization. Not all user-adaptive systems have adaptation components, but instead have the functionality mixed within other components. In many cases, such an implementation is sufficient.

Adaptation description. An adaptation description is a description of an adaptation model that in an appropriate language and depth such that it can be used by an adaptation engine to determine the adaptive personalization that is to be performed. This means that an adaptation description also concretely describes an adaptation model.

Adaptation engine. An adaptation engine is a generic adaptation component that can implement different adaptation models. An adaptation engine may either be embedded into a system or be a service that is provided for one or multiple independent systems.

Adaptation model. An adaptation model describes how the adaptive personalization in a user adaptive system is performed. It describes how events lead to user model updates and how the user model is used to give the information for personalization. While it does describe the personalization and how the information they need is to be retrieved, it does not describe how the personalization is to be performed. The latter is the responsibility of the application.

1.5 Thesis structure

The rest of this thesis is organised as follows. The outline is schematicaly depicted in Fig. 1.2.

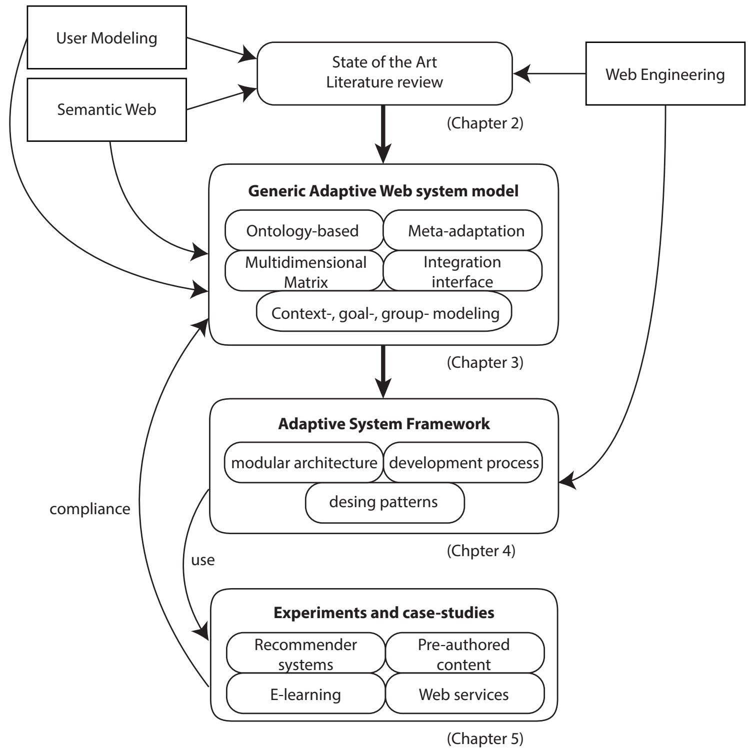


Figure 1.2: Thesis outline.

After the initial chapter, which gave a brief introduction to our work and the used terminology, in the second chapter we will focus on existing solutions and principles. We will explain the principles of hypermedia adaptation – adaptation and *User Modeling* techniques and describe some of the existing AHS reference models used to design adaptive web systems. We will also mention the basics of the *Semantic Web* technology which we believe to be the appropriate direction for the development of adaptive systems that can interchange user data easily.

The very essence of our research is described in the third chapter. First, we will discuss the important requirements for the current Adaptive Hypermedia Systems. The requirements originate from the state of the art analysis and outline the extensions provided by the proposed model. The formal description of the model will be presented. Finally, our model will be compared with AHAM, an influential, however outdated AHS model.

In the fourth chapter, we present the design of generic adaptive system components and adaptive system framework that can be used as foundation stone while developing various adaptive hypermedia systems. The framework is based on the theoretical model, extends the ideas with *Web Engineering* methods and provides the base implementation of adaptive hypermedia system.

Following the core research topic, we focus on evaluation of our results. In chapter five, we will demonstrate the experiments carried out to evaluate the theoretical results. The experiments constitute the necessary steps that were required to design the framework, or, some of the Web applications use the framework and were developed to verify its applicability for various types of adaptive systems. At the same time, our experiments show the compliance with the proposed adaptive web system reference model.

The last chapter concludes the theses, reviews results of our research and proposes directions of further research.

CHAPTER 2

Background and State of the Art

Many researchers have focused on the field of web adaptation and personalization during the last decade. Some adaptive systems and frameworks have been developed. In this chapter, we will summarize used approaches, advantages and disadvantages of existing systems and recognize the missing tasks that still need to be done in the area of adaptive hypermedia systems.

2.1 Adaptive hypermedia systems

With the growth and increasing availability of the World Wide Web, also the importance of hypermedia systems rises. Soon, the classical “one size fits all” approach didn’t comply with the increasing amount of hypermedia systems users. Users have different preferences, needs, experience, knowledge and wide range of other characteristics, which make them unique within the group of other users. However classical hypermedia systems offer the same environment for all the unique users. Such a system can offer at most a suitable compromise for all users.

The issue of inflexible hypermedia systems resulted in emerging of the adaptive hypermedia systems (AHS for short) research field.

Personalization and adaptation are not new techniques in the field of Computer Science. There are relevant issues in well-established research fields like Control Systems, Climatology, Biology, Evolutionary Research or System Theory. Personalization systems represent a specific subtype of general adaptation systems [73]. Therefore, personalization can be regarded as adaptation towards a specific user. The adaptive systems are mostly interactive systems where a user interacts with the system. Interaction causes invocation of events. An event results in some action modifying the state of the system.

Adaptive web systems belong to the class of user-adaptive systems [27]. The information about an individual user is stored in a user model. Based on the user model information, the adaptation effect is performed. The process of adaptation utilizes specialized modeling and adaptation techniques [22].

Most current adaptive hypermedia systems are based on common fundamentals. The adaptation usually depends upon basic components that are consistent with the reference models proposed in the beginning of the adaptive hypermedia research era like AHAM [54], Munich Model [108], FOHM [9] or GAHM [130]. There are three fundamental models: domain model, user model and adaptation model. As defined in [22], the adaptive hypermedia system should satisfy three criteria: it should be a hypertext or hypermedia system, it should have a user model, and it should be able to adapt the hypermedia using this model.

As described in [166] adaptive application must provide a domain model describing how the content of application is structured. Domain model represents the concepts and the concept groups of application information area. Concepts are interconnected by concept relations. Adaptive system must construct and maintain a fine-grained user model that represents user's preferences, knowledge, goals, navigation history and other important aspects. Adaptive system must be able to adapt the presentation. This functionality is provided by an adaptation model (in AHAM called teaching model) that performs adaptation usually based on predefined rules.

Adaptation model represents the adaptive process performed by the system. Adaptation model is usually represented by rules defined over the concepts and their relations.

These fundamental components of an adaptive system are usually extended with additional models to be able to capture all concerns of the adaptation process. Context model is used to bring the adaptation to a particular context of the user. That includes place, time, type of device being used, etc.

2.1.1 Application areas

Adaptive hypermedia techniques can help solve identified problems in various application areas. Brusilovsky [22] has named six kinds of hypermedia systems where research projects have been applied. These are: educational hypermedia, on-line information systems, on-line help systems, information retrieval hypermedia systems, institutional information systems, and systems for managing personalized views. We will review each of these application areas in the context of current research on adaptive hypermedia.

2.1.1.1 Adaptive learning

E-learning has been defined as *"...just-in time education integrated with high-velocity value chains. It is the delivery of individualized, comprehensive, dynamic learning content in real time, aiding the development of communities of knowledge linking learners and practitioners with experts"* [154]. Efforts have been made to define standards for Learning Management Systems (LMS). Three main metadata standards are IEEE Learning Object Metadata (LOM), Dublin core metadata and SCORM metadata [90].

Adaptive environments are very closely related to the area of e-Learning. Educational applications are developed as tools for teachers and students and help the users to manage the learning materials. The problem of typical e-learning applications is that they do not take into account different preferences and knowledge of the users. This issue was the initial motivation for the adaptive systems research [22]. The adaptive web-based courses offer obvious advantages for students by allowing fast access to relevant educational resources at any time or place.

Adaptive hypermedia is very suitable for online syllabi and courses. Students differ in knowledge, learning styles, each one has different speed of learning or each would prefer a different path through the curriculum. Therefore, adaptive courses are the most often use cases of adaptive hypermedia. Some adaptive techniques have been designed specifically for adaptive learning systems.

Among e-learning applications, we can find monolith LMS, such as Blackboard or WebCT, which provide various activities. In the past years of adaptive hypermedia research, many adaptive e-learning applications have been developed. We can name some examples as AHA! [53], Interbook [25], ELM-ART [163], KBS Hyperbook [91], APeLS [44], etc.

Current trend heads towards integration and reusability. This can be achieved by the means of Semantic web [151].

2.1.1.2 On-line information systems

The goal of on-line information systems is to provide a reference access to information. This is different to educational hypermedia, where systematic approach and stepwise introduction to the problematic are assumed. As information systems we can categorize a variety of applications, eg. libraries, city guides or online encyclopedias. All these applications can gain great benefits by the personalization to the needs of individual user. Depending on the targeted application domain, their hyperspace can range from reasonably small to very large.

Nodes of the hyperspace are usually represented by concepts and concepts consist of several pages. In some cases, user knows which concepts of the information space he needs to access. However, in many cases, users need navigation support. The requirements depend on user's goal, and the system has to know it to be able to give navigation support or find relevant pieces of information. However, to identify the user's goal is a difficult task. It often happens, that the user has even more simultaneous goals. User goals are usually predefined in taxonomies, but they are usually very specific for pre-defined domains, or highly abstract, based on human psychological motivations [138]. To reuse existing goals, innovative approaches have to be used.

2.1.1.3 On-line help systems

On-line help systems are very close to on-line information systems. The difference is, that on-line help systems are tied together with a particular application (office application, development environment, expert system, etc.). Help systems suffer with the same issue as other on-line information systems – serving different information to different users is needed. The information space is usually small in this type of applications, and we are able to identify user's context better. Systems that are aware of the context form which the user called for on-line help are called context-sensitive help. Therefore, the adaptation is not as important as in systems with large hyperspace, however, adapting the help content, presentation and links to related topics can significantly improve user experience.

2.1.1.4 Information retrieval hypermedia

Combination of traditional information retrieval (IR) techniques with a hypertext-like access from the index terms to documents is used in information retrieval hypermedia systems. Adaptive IR hypermedia systems are using similarity links to provide the possibility of browsing the hyperspace. The challenging problem is to support information retrieval tasks in the unrestricted web hyperspace. This hyperspace cannot be structured "by hand" and the links between documents cannot be provided by a designer. Links are generated by the system, usually using similarity measurements.

2.1.1.5 Institutional information systems

Institutional information systems are more work-oriented and differ from more search-oriented on-line information systems in the size of the working area of a user. Institutional information systems support work of some institution. These systems can join a set of databases into a single, reasonably large, hyperspace. Institution employees use the information system for everyday work, and according to their profession, they may need an access to a very small subset of it. A lot of navigation possibilities can distract users, and they usually never need access to parts of information space, which does not correspond to their scope of work.

Another problem solved by personalization in institutional information systems is the help for new users. New employees are not familiar with the structure of hyperspace and can get lost easily. This is similar to the problem solved by educational hypermedia.

2.1.1.6 Systems for managing personalized views

The World Wide Web provides access to various sources of information and offers miscellaneous services. Users, however, need and regularly use only a subset of these services and need access only to a subset of the information space. To solve this issue, its practical for users, to create personalized views on the entire hyperspace that would protect them from the complexity of the overall hyperspace. This solution is similar to the institutional hypermedia. However, in open web, we face new dimension of the problem – the dynamical character of hyperspace. Items can appear, disappear, or evolve. Personalized views need continuous management. From this point of view, personalized views are similar to IR systems. We need to search for new, relevant items and identify the outdated or non-relevant parts. Adaptation of personalized views can be based on user's goals, preferences, background and other characteristics related to the view purpose.

2.1.1.7 Other application areas

In the recent years, we can see application of adaptation and personalization in social networks [4] and various communication channels [88]. Current trend of mobile devices is also a good perspective for utilizing adaptive personalization [93]. Modern approaches of Web 2.0 allow focusing not only on mobile devices, but on service-oriented architecture in general [135, 64].

Web applications are the mainstream of software development today, and personalization penetrates through all areas. Since Brusilovsky classified adaptive systems into the described six categories, hypermedia applications had evolved. New application areas and technologies have emerged. However, since even the original categories overlap, we can usually match the characteristics of the employed adaptation to the mentioned categories. New application areas of AHS are constantly appearing. Therefore, it is difficult to capture the whole scope of systems, where personalization is applied.

2.2 Adaptation methods and techniques

We are focusing on adaptive hypermedia, so the question is: *“**What** can be adapted in adaptive hypermedia?”* At some level of generalization, hypermedia consists of a set of nodes (concepts) or hyperdocuments connected by links (concept relationships). Concepts and their relationships are represented in a *domain model* and usually form a hierarchy. In our work, we assume not only the data, but also metadata and semantics provided by ontologies. Adapted can be the content of regular pages (content-level adaptation) and the links from regular pages, index pages, and maps (link level adaptation). The first one is also called adaptive presentation and the second one adaptive navigation support [22]. Mostly, particular selected methods from both areas are used.

Related question to the previous one is: *“**How** is the adaptation performed?”* There are several approaches that can be used to personalize the information presented to the user. Methods represent generalizations of a technique. Method is based on a clear idea which can be presented at the conceptual level. Each method can be implemented by different techniques, and some techniques are used to implement several methods using the same knowledge representation.

The term “adaptive presentation” was used by Brusilovsky as a synonym to content-level adaptation. In later research [102], it became obvious, that adaptive presentation needs to be considered separately. Some forms of content adaptation, that were defined by Brusilovsky, only change the presentation. The same way, some forms of adaptive navigation techniques do not change possible navigation, but only change “suggestions” by changing the presentation.

The up-to-date taxonomy of adaptation techniques (Fig. 2.1) distinguishes three types of techniques that are partially overlapping – content adaptation, adaptive presentation and adaptive navigation.

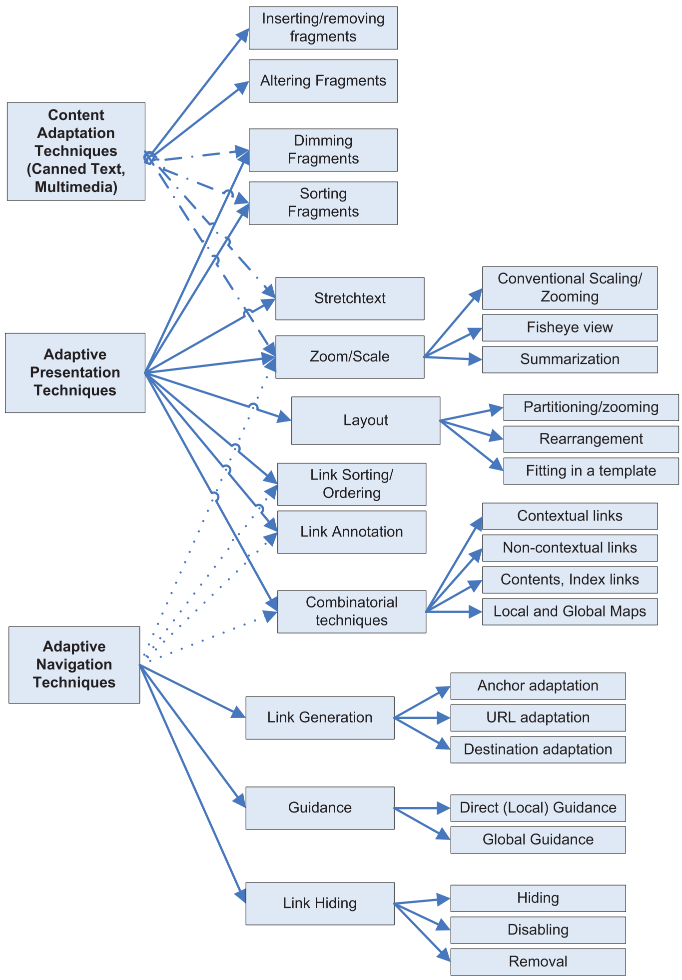


Figure 2.1: Taxonomy of adaptation techniques [102].

We will give a brief overview of the particular methods.

2.2.1 Content adaptation methods

Additional Explanations In adapted document, the basic content of the page is extended with particular information. Additional information is shown with respect to user's knowledge.

Prerequisite Explanations With appliance of the user model, the system checks if the user's knowledge complies with all prerequisites. The explanation of all missing prerequisites is added to the document to facilitate the comprehension of the concept.

Comparative Explanations New concepts are explained in the way of linking them with some concepts already known to the user. Typically, the similarities and differences between these two concepts are pointed out.

Sorting Explanations Parts of the document are sorted by the relevance of information to the objectives of the user. Level of his knowledge, possibly some other characteristics are taken into account.

Explanation Variants In the source document, there is defined a number of different explanations of given concept. In the final document, the most suitable variant for the user is shown.

2.2.2 Adaptive navigation methods

Direct Guidance In this method, the user is sequentially guided through the hyperspace. The "next button" is offered to the user. The system selects the best page suitable for the user in compliance with the information stored in the user model. Another possible technique is to choose a sequence of pages. This sequence should help the users to find their paths in hyperspace.

Adaptive Sorting The links within the hypermedia document are sorted according to importance for the user. Sorting of the links can be made by the similarity of the links in current document. Another way of sorting is made by the prerequisites. This method is similar with the one used for adaptive presentation. The only difference is that instead of showing corresponding text, links with needed information are labeled with higher relevance.

Adaptive Hiding This method excludes the possibility of visiting pages with no relevant information. Links leading to such pages are hidden or disabled. This restriction avoids the user to be lost in a large hyperspace.

Links Annotation In this method, the links are annotated to show some information about the content of the pages to the user. The annotation can be provided in a textual form. Another way is to use some visual augmentation, e.g. relevance is represented by a colored sign next to the link.

2.3 Modeling techniques

User modeling in adaptive systems is described by the *adaptation model*. This model defines how adaptation of domain model concepts is performed based on the information stored in a user model, and how the information in a user model is updated.

We will complete the questions from section 2.2 with some related with user modeling. Following questions are very close: *"**Where** should we apply the adaptation?"* and *"**When** should we apply the adaptation?"* There are two possible conceptions of these questions. Either we ask about the application in which the adaptation is applied, or we mean the environment in which the adaptive application is used. The first case was captured in the Brusilovsky's classification of application areas (as was mentioned in section 2.1.1). The second case is related to the adaptation model, where the questions correspond to specific dimensions of rules and constraints and are bound up to the context awareness of the application.

Next important question we ask is: *"**Why** do we adapt?"* The answer to this question is related to user's objectives and goals. Although, the goals can be stored in a similar way as other user properties, current trends in the design of adaptive systems head towards proving more sophisticated external goal model.

The last question is: *"**To What** can we adapt?"* The *user model* is a representation of information about individual user that is essential for an adaptive system to provide the adaptation effect, i.e., to behave differently for different users [28]. According to Seleman [149], we can classify user models along three layers: what is being modeled, how this information is represented and how different kinds of models are maintained.

2.3.1 Characteristics of users

The first user model classification layer is: *"What is being modeled in adaptive hypermedia?"* The user model represents features of the user and sometimes also the current context of the user's work. Important user features to be modeled are the *user's knowledge, user's interests, user's goals and cognitive styles.*

The most important user feature is the user's knowledge of subject being taught or the domain. However, this feature is changeable, and we have to take into account both increase (learning) and decrease (forgetting).

Another important user feature to be modeled are the user interests. They have always been important in filtering and recommender systems. However, due to rapid growth of the volume of information in adaptive hypermedia systems, the user interests are becoming important in these systems too.

The most changeable user feature is the user's goal. Depending on the application, it can be the goal of the work, an immediate information need or a learning goal. Recognizing the user's goal is a difficult task, especially in web-based adaptive systems, where the flow of information from the user is thinner than in traditional desktop systems. To determine the user goal, user can be asked to select it, or some systems are able to determine it gradually. Some systems show the assumed goal to the user in the spirit of a glass-box adaptation, and the user is allowed to change it.

User's previous experience outside the domain of a specific system is being modeled as user's background. The system can, for example, display content to the users depending on their knowledge of medical terminology, or depending on whether they are native or non-native speakers.

Features, that together define a user as an individual, are called individual traits. For determining these features, mostly psychological approaches are used. They include two groups of traits – cognitive styles (preferred approach to organizing and representing information) and learning styles (the way people prefer to learn).

Evolution of hypermedia systems made it necessary to extend the characteristics of the user to different features as well. Kobsa [105] suggests to distinguish between *user data, usage data and environment data*. User data represents the user characteristics as was explained above. Usage data describe the user's interaction with the system (i.e. time spent on a page, ratings, purchases, etc.). Environmental data can be related to the software environment (browser, plugins, etc.), hardware environment (device type) or user's environment provided by various sensors (location).

A new research direction within adaptive hypermedia systems is the context of the user's work. This interest was caused by increasing popularity of mobile devices and ubiquitous systems. The context features can include, for example, user's location, user's platform (hardware, software, network bandwidth), physical conditions (light, temperature), social context, etc.

2.3.2 Classification of user models

The second question asked when designing a user model in an adaptive hypermedia system is: *"How is the information about the user represented?"*. The question can be answered by reviewing and classifying user model architectures.

The simplest form of a user's knowledge model is the scalar model. This model estimates the level of user knowledge by a single value on some scale (e.g. a number ranging from 0 to 5). The shortcoming is its low precision. For that reason, many adaptive hypermedia systems (AHS) use various kinds of structural models, which attempt to represent user knowledge of different fragments of the domain knowledge independently. The most popular form of a structural model is an overlay model. This model represents an individual user's knowledge as a subset of the domain model, which reflects the expert-level knowledge of the subject. An example of a student's overlay model is shown in Fig. 2.2. This example was adopted from the NetCoach course authoring system [162].

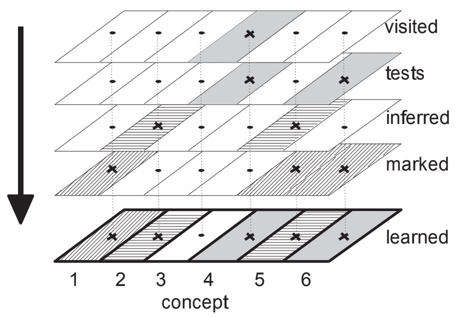


Figure 2.2: Example of a student's overlay model [164].

The overlay models have often been criticized for being "too simple," because the state of user knowledge is never an exact subset of expert knowledge. To model user misconceptions, an overlay model was expanded into a bug model. One of the forms of the bug model is called the perturbation model. It assumes that several incorrect perturbations can exist for each element of domain knowledge. An even richer model makes possible to reflect the development of user knowledge from the simple to the complex and from the specific to the general. Such model is known as a genetic model.

As it was mentioned, there are more powerful models than the traditional overlay model, but they are also much harder to develop. Therefore, the practical use of these models has been quite limited. On the other hand, overlay models are extremely popular and almost every web-based AHS is based on some form of the overlay model.

2.3.3 User model maintenance

After the user model is designed and populated with data, we may ask: *"How different kinds of models are maintained?* User models are constructed from information sources using a variety of construction techniques based on machine learning or information retrieval. The information profiles could be constructed manually by experts, but it would be too difficult for users. Automatic techniques are more appropriate. Some authors warn against fully automatic updates [150]. User feedback, which requires minimal effort, should be used.

For the particular methods used for creating user models, please refer to [75].

2.4 User feedback

A very important aspect of user adaptive systems is the user feedback. User provides by means of user feedback necessary information about the actual level of his characteristics (eg. knowledge of topic, goal or preferences). Fig. 2.3 depicts the user feedback as part of the adaptive system.

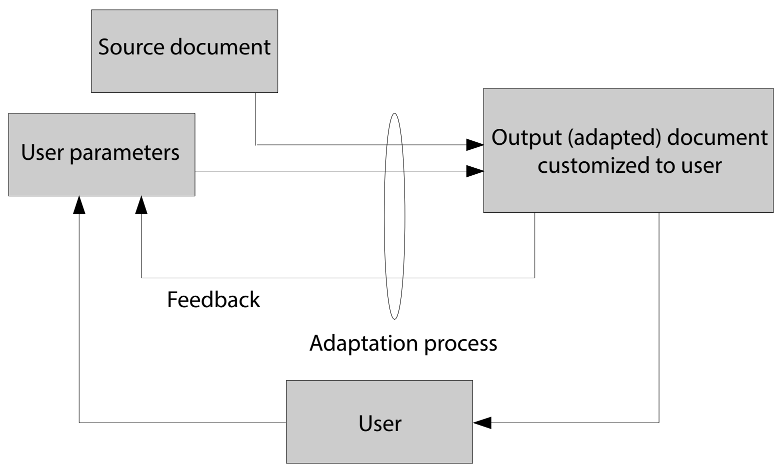


Figure 2.3: Principle of an adaptive hypermedia system [31].

There are different situations, where the system can obtain user's preferences by the feedback. We distinguish three situations:

- Information is requested before the start of the adaptation process. To request information various questionnaires, or preference screens can be used. Advantage of this approach is, that the system has some information about the user, and immediate adaptation is possible. This type of user feedback is one of the methods solving so called "cold start" problem. However, disadvantage of this approach is, that the user is bothered and burden by filling the requested information.

- Information is obtained within the operation of the adaptive system. This type of feedback can be further divided by the involvement of the user. We can distinguish implicit and explicit feedback.

  - **Implicit user feedback** is performed without the user knowing. The application observes user's actions. In this case, user performs actions and provides feedback data the same way as he would do in a non-adaptive system. The observed information could include which pages has the user visited, how long was he reading some page or how many times has he returned to view some document.
  - **Explicit user feedback** requires additional user actions. Information about user is acquired by explicitly asking the user, by requiring to fill a form or by actions directly related to the adaptation process. User can be asked by explicit question (e.g. Do you need more information?). More frequent, especially in adaptive learning systems, is a button or icon (e.g. I don't understand). The most intrusive is asking a user for information by a complex form. This includes acquiring information before the adaptation is started (new user). In case of retrieving the information from another system (a system that the user used before), we do not distinguish if the information was obtained by implicit or explicit feedback.

- Information is obtained after some time of using the adaptive system. In this case, data-mining techniques are used. This involves processing of a large amount of data using statistical method and artificial intelligence algorithms. The goal of this process is to extract information from a data set and transform it into an understandable structure for further use – particularly, a user model is populated with new observations about the user, that can be used to perform the adaptation. More details about data mining in the area of adaptive hypermedia systems can be found in [158].

2.5 Existing Reference Models

Already in the beginning of the AHS era, researchers realized the need of structural design for adaptive hypermedia. Several models were designed for Personalized Web. We will describe the most important of them in next paragraphs.

2.5.1 AHAM

Adaptive Hypermedia Application Model (AHAM) [54] is one of the first and most influential formal models for adaptive hypermedia. It is an extension of Dexter model [87], a widely used reference model for hypermedia. The adaptation is based on a *Domain Model*, a *User Model* and a *Teaching Model* which consists of pedagogical rules (Fig. 2.4). The AHAM originally comes from the field of educational hypermedia, and these origins can still be found in the model.

Like the Dexter model, AHAM has three layers – the *Run-time Layer*, the *Storage Layer* and the *Within-Component Layer*, connected by the interfaces *Presentation Specifications* and *Anchoring*. The Domain Model uses concept components for abstract representation of information. A component's information consists of a set of attribute-value pairs, a sequence of anchors and a presentation specification. The User Model is used to store the information about one specific user. It is an overlay of the Domain Model. The Teaching Model uses rules to define how the Domain Model and the User Model are combined to provide ways to perform the actual adaptation.

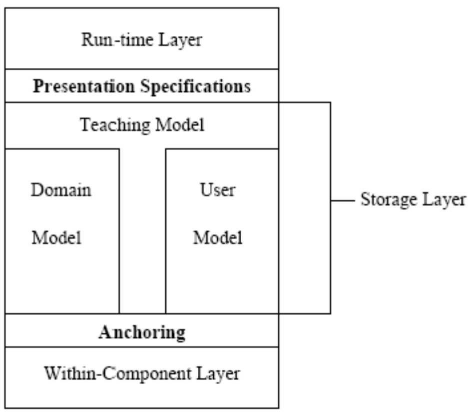


Figure 2.4: The AHAM model [54].

The actual adaptation is performed by an adaptive engine. The adaptation is based on rules. The rules can either be based on the concept relationships in the Domain Model, or specific rules are bound to specific concepts or groups of concepts. The disadvantage of rule-only-based adaptation has been tried to be eliminated by post and pre concept access rule execution, but it still does not eliminate the problem [60].

2.5.2 Munich Reference Model

The Munich Reference Model [108] is also an extension of the Dexter Model and in a similar way adds a User Model and 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.

2.5.3 XAHM

The XML Adaptive Hypermedia Model (XAHM) [32] is defined in two logical levels. The upper level is a graph-based layered model for the description of the logical structure of the hypermedia. The lower level is composed of XML-based models for the description of both the metadata about basic information fragments and the "neutral" pages to be adapted.

2.5.4 FOHM

The Fundamental Open Hypermedia Model (FOHM) [9] is based on the prior work with Fundamental Hypermedia Protocol (OHP) which provides a reference model and architecture for Open Hypermedia systems. It provides the facility to attach context and behavior objects to the model at various locations. An engine, Auld Linky, is required to instantiate and process the model.

2.5.5 GAHM

The Goldsmiths Adaptive Hypermedia Model (GAHM) [130] is an abstract model. It takes a formal approach to the modeling of adaptive hypermedia. The model consists of three groups of functions. The functions in H-Region model non-personalizable hypermedia-based interaction, in P-Region user-initiated tailoring and in A-Region system-initiated tailoring of hypermedia content.

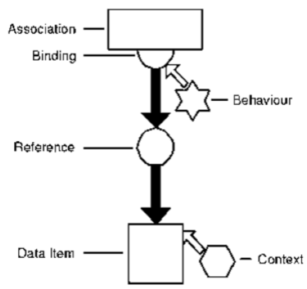


Figure 2.5: The Structure of a FOHM Object [9].

2.5.6 GAM

The Generic Adaptivity Model (GAM) [60] is a state-machine based model, which does not restrict only to hypermedia, but can be used as the basis for adaptation in all kinds of applications. Compared to AHAM, the GAM is more low-level and does not provide hypermedia specific concepts. On the other hand, GAM provides an explicit interface model and the concepts of push and pull adaptation.

The model assumes that these user-adaptive systems are also interactive systems, i.e. systems where a user interacts with the system. Each interaction generates an event. The event results in an action which causes a state change.

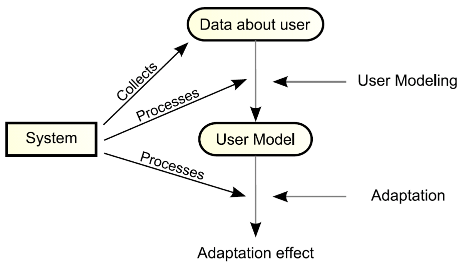


Figure 2.6: Classic user modeling loop for adaptive systems [22].

In [22], Brusilovsky presented a graphical model for user modeling in adaptive systems. The model is given in Figure 2.6. In [61], de Vrieze presented a more accurate model (Figure 2.7). He states that some answers are not included directly in the user model, but can be better calculated on demand. The reasoning that transforms user properties into answers about the user is put together with the actual changing of the application interface.

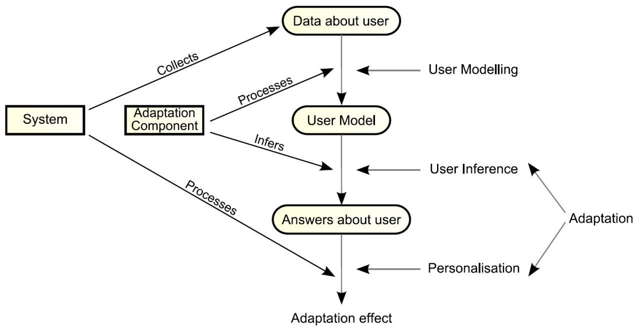


Figure 2.7: Improved user modeling loop for adaptive systems [61].

The GAM model is based on this principle. The two parts of the adaptation system are push adaptation and pull adaptation. The properties of systems based on each of these adaptation types or their combinations are described using the two-dimensional classification framework in [59]. The push adaptation represents updating of the user model and pull adaptation represents querying of the user model.

Unlike the Munich reference model, the GAM aims to ensure confluence and termination, which are important properties for accuracy of the model. The GAM model is more suited as a description for generic adaptivity than the AHAM model is. However, in the area of adaptive hypermedia, we are currently still missing a model that on top of GAM can provide specific hypermedia concepts. We will try to provide such extension in our work.

2.5.7 LAOS

The Layered Model for adaptive hypermedia authoring LAOS [48] extends AHAM by adding an additional layer between the Domain Model and the User- or the Adaptation Model. This additional layer provides goals to give a focused presentation and constraints to limit the space of the search. Furthermore, LAOS's Adaptation Model is different from that of AHAM. It is based on LAG, the three-level granularity model [46]. The *low-level adaptation* is similar to AHAM, defines traditional adaptation techniques and is based on IF-THEN format rules. The *medium-level adaptation* provides higher level of adaptation rules semantics and defines the adaptation language as a wrapper for adaptation rules. The *high-level adaptation* is the most advanced layer which provides support to define various adaptation strategies.

2.5.8 Social LAOS

The Social LAOS model [78] is an extension of LAOS model and adds an additional layer – the Social Layer. This new layer is oriented vertically, as it affects all other layers directly. Social LAOS extends its predecessor with social annotation and users' collaborative activities. It allows users to contribute in the authoring phase to provide more valuable adaptive content.

2.5.9 GAF

The Generic Adaptation Framework (GAF) [104] aims to provide a reference model that encompasses the most general architectures of the present and proposed future architectures. Compared to AHAM, the model adds search, group and higher-order adaptations, enhanced reasoning and data mining support, open corpus adaptation, context-awareness and allows domain structure definition by an ontology. Furthermore, the author introduces the Generic Adaptation Process (GAP) which defines the interaction in AHS.

2.6 Existing adaptive systems

Based on the models presented in the previous section, adaptive systems were developed. We will focus on the most important systems for the research area and compare their features.

Based on the AHAM model, Adaptive learning system called Adaptive Hypermedia for all (AHA!) [53] has been developed. But AHA! was not a full implementation of the reference model, as AHAM was much more general. AHA! was developed on the Eindhoven University of Technology. It is designed for the development and presentation of online courses and tutorials. The system allows using simple adaptation techniques inside the hypermedia document. These techniques are used for the content adaptation as well as the link adaptation. The AHA! system also allows to design tests to verify the student's knowledge of the topic.

Interbook [21] was created as a tool for conversion of static electronic textbooks into adaptive learning environments. Compared to AHA!, Interbook does not support adaptive presentation, but it only supports adaptation of links. Interbook is system-dependent and available for Apple Macintosh systems only. The main view of the system is divided into glossary window and textbook window. Links to required concepts and their prerequisites are provided.

KBS Hyperbook [92] was created as an adaptive introductory course on computer science. In contrast to AHA! and Interbook, this system does not monitor, which pages the user has visited. The reason behind this decision was, that it is difficult to measure the knowledge of a user who reads the page. Instead, the system is based on a goal-driven approach. The system reflects, how the student performed in the assigned project. The gained knowledge of the user is used while selecting a new project and also when some prerequisite knowledge is missing. To support user in case of lack of prerequisite knowledge, sequence of units containing the needed information is generated.

There are few attempts to develop libraries and frameworks supporting adaptation. Generic Responsive Adaptive Personalized Learning Environment (GRAPPLE) [55] has been developed at the Eindhoven University of Technology, as a part of the FP7 project[1]. This system is focused on adaptive learning. It is integrated with existing learning systems (e.g., Blackboard[2] or Moodle[3]). The most important contribution of this project lies in integrating the adaptive delivery of the teaching materials for the course into a supported learning process.

Another project, specifically focused on adaptive learning, is the Adaptive eLearning Platform [12]. This system is the implementation of the Virtual Apparatus Framework, a content development paradigm modeled after the process of developing a teaching lab activity. The approach used in this project is tightly connected to the learning process and is based more on pedagogical principles than on software engineering.

A further promising project is also HyperAdapt [124]. In this project, a specialized approach utilizing an aspect-oriented programming is used. The authors place the adaptivity into separate modules called adaptation aspects. The aspects are not applied on a model level, but on XML documents.

One of the solutions intended to extend legacy web applications with adaptive behavior is the Adaptive Server Framework [80]. Compared to our solution, this project is focused on server-side components only. The design principle is to separate the implementation of adaptive behavior from the server application business logic. The coupling of components is ensured by a message-based communication.

A similar solution is the Rainbow project [43]. This project uses an architecture-based approach. The system adaptation is predefined by the architecture style of the system.

[1] GRAPPLE Project – `http://www.grapple-project.org`

[2] Blackboard Learning System – `http://www.blackboard.com`

[3] Moodle Learning System – `https://moodle.org`

The commonly used design principle is the principle of a modular architecture.

Another solution is the MUSE semantic framework [37]. The framework is built on multidimensional ontological planes. The intersection between the planes allows the representation of semantic rules. A similar principle is used in GOMAWE, where a multidimensional matrix of rules is used to infer the information not explicitly stored in the user model; see Chapter 3.

Adaptive systems are usually based on a reference model, but do not implement all of its features. Such systems were usually developed as a prototype in a laboratory environment. Prototypes developed at universities usually focus on adaptive learning, which is right opportunity for evaluations within classes.

2.7 Development methodologies of AHS

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) [74], Relationship Management Methodology (RMM) [98], Enhanced Object-Relationship Model (EORM) [111] and Web Site Design Method (WSDM) [57]. An Overview of additional and more recent development methodologies for software and Web engineering can be found in [6, 156]. 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. 2.8 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 2.8: Typical phases of a software devel. process.

Model-Driven Architecture (MDA) [118] 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 [133]. For object-oriented software systems modeling, we have a standard, widely-adopted, formally defined language – UML [18]. 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 [108] is an extension of the Dexter model. It was proposed in the same period as the well-known Adaptive Hypermedia Application Model (AHAM) 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) [142] 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 [11].

Another method to specify design of complex Web sites is WebML [42]. 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 [6].

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 [101]. This is achieved by the UML stereotypes.

Table 2.1 provides the mappings for the most important constructs.

Table 2.1: Basic Profile Constructs [101].

OWL Ontology Item	**UML Construct**	**UML Stereotype**
`Ontology`	`package`	`ontology`
`Class`	`class`	`ontologyClass`
`Relation`	`class`	`relation`
`Individual`	`class`	`individual`
	`association`	`individualOf`
	`association`	`typeOf`
`ImportOntology`	`dependency`	`<none used>`

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 [45]. The adaptive-system development process can be divided into four phases [115]: Conceptual Phase, Presentation Phase, Navigation Phase and Learning Phase (Fig. 2.9).

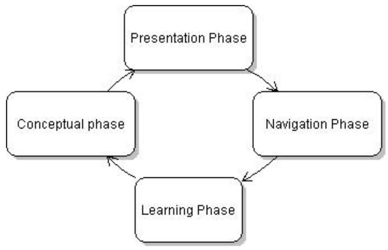


Figure 2.9: Course design phases

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.

2.8 Adaptive System Evaluation

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 [120]. It is important to use an appropriate method for evaluation [77]. Evaluation should ensure savings in terms of time and cost, completeness of system functionality, minimizing required repair efforts, and improving user satisfaction [125]. 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.

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 [66]. 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 [134]. 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 data strictly related to the performed task. Adaptive systems, however, require much more information. This information might not be for the task required 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. Uses 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 [76].

2.8.1 Evaluation methodologies

Recent research has identified the importance of user-adaptive systems evaluation. Reviews on the topic have been published by several researchers [76, 159, 120, 3]. 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.

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 [94]. 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 [3].

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 [165]. However, in recent studies, the importance of this approach is pointed out [134]. This method of evaluation is derived from empirical science and cognitive and experimental psychology [76]. 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 [29] 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 [120]. 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 [159].

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.

2.9 The Semantic Web and its role in user modeling

Semantic Web technologies have begun to creep into use in the hypermedia applications. Information is represented in a way such that machines can use it for automation, integration and reuse of knowledge across applications. The machine understandable contents are called metadata and their semantics can be specified using ontologies. Ontologies play an important role in the Semantic Web as they provide a common shared model to represent a domain and to reason about the objects in the domain.

2.9.1 Ontologies and the Semantic Web stack

Semantic Web is the new evolvement of the web, where content with formal semantics is supported. The idea of machine understandable content consumed by intelligent agents was presented by Tim Bernes-Lee [15]. To express the semantics, the knowledge space can be represented using domain ontologies.

There are many definitions of the term ontology. The term is borrowed from philosophy, where ontology is a systematic account of Existence.

In computer science, ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts. It is used to reason about the objects within that domain. Ontology is an explicit specification of a conceptualization. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and formal axioms that constrain the interpretation and well-formed use of these terms [83]. Following are other definitions of an ontology. Fensel in [67] defines ontology as “a shared and common understanding of a domain that can be communicated between people and heterogeneous and distributed systems”. Sowa [152] takes ontology as “a catalog of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses a language L for the purpose of talking about D”. Simple definition is by Huhns [97] saying that ontology is “a computer model of some portion of the world”. Generally, an ontology can be defined as a linguistic artifact that defines a shared vocabulary of basic concepts for discourse about a piece of reality (subject domain) and specifies what precisely those concepts mean [99].

In Figure 2.10: you can see the schematic representation of semantic web. On the left side, there is the static part concerning languages. Basis of the hierarchy rep-

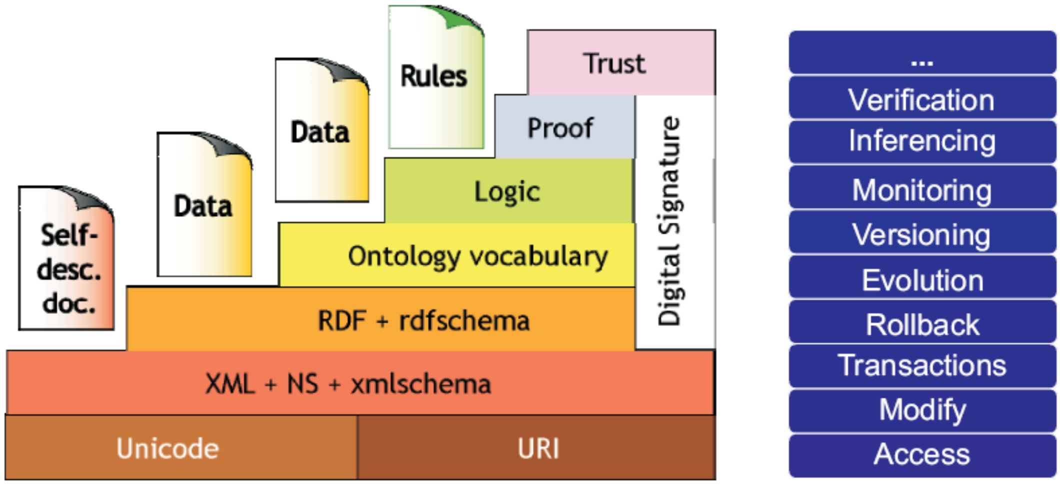


Figure 2.10: Static and dynamic aspects of the Semantic Web layer cake [129].

resent the Unicode, URI and namespaces (NS). The XML[4] is used for the purpose of content storage. The language is extensible for any domain-specific content. The structure of a document can be described by the DTD or XML Schema[5]. This ability of a structure description contributes significantly in guaranteeing that the content is machine-readable.

XML has been further extended by the Resource Description Framework (RDF)[6]. The RDF metadata model is based upon the idea of making statements about resources in the form of subject-predicate-object expressions, called triples in RDF terminology. This allows making simple assertions about web resources or any other entity that can be named. RDF Schema[7] extends RDF by class and property hierarchies that enable the creation of simple ontologies.

The ontology layer can be represented by the Web Ontology Language (OWL)[8] which is a family of richer ontology languages that augment RDF Schema. OWL adds relations between classes (disjointness), cardinality (exactly one), equality, richer typing of properties, characteristics of properties (symmetry) and enumerated classes. OWL has three variants: OWL Lite, OWL DL, and OWL Full (ordered by increasing ex-

[4]XML – http://www.w3.org/XML/
[5]XML Schema – http://www.w3.org/XML/Schema
[6]RDF – http://www.xml.com/xml/pub/98/06/rdf.html
[7]RDF Schema – http://www.w3.org/TR/PR-rdf-schema/
[8]OWL Web Ontology Language Semantics and Abstract Syntax – http://www.w3.org/TR/owl-semantics/

pressiveness). OWL Full is the most expressive variant of OWL. It was designed to preserve compatibility with RDF Schema. OWL Full is undecidable. Currently, no complete implementation of OWL Full exists. OWL DL is a decidable sublanguage of OWL Full. It is based on the $\mathcal{SHOIN}^{(\mathcal{D})}$ description logic. OWL Lite was defined with the aim of more tractable inference problems and easier to understand. However, due to the similar difficulty of tools' development compared to OWL DL, OWL Lite is not widely used.

The logic layer will provide languages, which will be used to make deductions and to derive new information from existing data. The proof layer will describe the steps taken to reach conclusions from the facts. Then it will be possible to pass these proofs around and verify them. The Semantic Web's vision is that once all these layers are in place, we will have a system in which we can place trust that the data we are seeing, the deductions we are making, and the claims we are receiving have some value. The goal is to make a user's life easier by the aggregation and creation of new, trusted information over the Web [65].

On the right side of Figure 2.10 are the aspects of the dynamic part of semantic web. These are addressed not so often as the static part, however, they are inevitable for putting the semantic web into practice [129]. Transactions and rollbacks of semantic web data should be possible. Evolution and versioning are important as well, because ontologies are usually subject to change [155].

2.9.2 Survey of application of Semantic Web technologies in AH

First, we should ask the question: "What can the Semantic Web bring to adaptive hypermedia that AH doesn't already have?". Cristea [47] answers this question within the scope of adaptation scaling and personalization between different systems. The important characteristic is the flexibility. The AH authoring should make use of semantically labeled reusable material.

Semantic Web technologies are becoming popular in the field of adaptive hypermedia, because they provide means to overcome the interoperability problems connected with current adaptive systems. Improved solutions will be based on ontologies and also take into account existing standards.

Research projects making use of ontologies in the adaptive hypermedia have emerged in all areas where classic adaptive hypermedia have been applied. Adaptive systems are very often used in e-learning, information systems, online libraries, e-commerce and other areas, where customized content or navigation is beneficial for the user.

Adaptive hypermedia systems have been modeling user characteristics in a closed information space. Modeling the user in a well-known domain is much easier. On the other hand, this approach makes the reuse of components and user model information very complicated.

Things change if we assume an open information space. Such information space can be defined by an ontology. Information can be shared between applications. In a such distributed environment, agents can be used to allow a modularized approach. Agents are then able to communicate using a prescribed ontology [8].

The user usually works with diverse services on the Web. Sometime, even the domain is quite similar. In other cases, there can be connections with other people having corresponding interests and new links to reusable data. On the web, many items are related. Though, adding semantics to the raw data brings new possibilities how the data can be used. Through semantic annotations, the vision of Linked Data [17] is becoming true.

In [41], researchers utilized ontology structure to solve cold start and diversity problems. The cold start problem [143] happens at the beginning of the interaction when the system does not have enough user data to provide appropriate adaptation. The diversity problem [131] occurs when recommendation results, although similar to the initial object are also very close to each other, thus lacking diversity and not providing the user with the satisfying alternatives.

AH systems are based on browsing and typically do not involve user queries. However, user feedback is an important element of AHS architecture. AH use metadata, concepts and conceptual relationships to drive their adaptation [153].

Recent research in the educational technologies focuses on the use of ontology and metadata. Semantic Web technology has not been extensively used for educational applications and therefore, there is a need to apply this promising technology in the field of e-learning. Many researchers are working on new approaches and applications [36], [63], [90]. Recently, ontologies have been used in adaptive learning to achieve user model integration, learning objects reuse and to devise appropriate content better for the user based on annotations. Semantic technologies have been used in projects like Personal Reader [62] and Edutella [122].

Along with the classic AHS involving navigation adaptation, items' recommendation makes use of the retrieval process. Ontology-based recommenders [116] [33] take advantage of the enhanced semantic representation.

There are many more application areas of the Adaptive Semantic Web. Ubiquitous information systems [39] make use of Linked Data in a distributed environment of mobile devices. From the machine-processable data benefit applications for e-commerce [95], e-government [113], or various information retrieval hypermedia systems [79].

Table 2.2 summarizes the advantages and disadvantages of Adaptive Semantic Web technology.

Table 2.2: Evaluating Adaptive Semantic Web technology.

Advantages	**Disadvantages**
Formal semantic	Need to use widely accepted ontologies
Possible integration of user models	Reasoning can be time consuming
Supports data reusability	Merging of vocabularies is necessary
Enables sharing resources	Trusted information has to be guaranteed

2.9.2.1 User identification

To exchange the user model information effectively among multiple systems, we need to be able to reliable identify the user. Information exchange can be very helpful to solve the cold start problem [126]. One of the means of identifying the user in distributed environment is "OpenID". This is an open, decentralized, free framework for verifying users' online identity [35]. Open protocols like OAuth are being used for secure user authentication. Such user identification and authentication is often used in combination with social networks. This way, we can integrate users and their data across multiple applications.

2.9.2.2 Integration of user models

Traditional adaptive hypermedia applications were closed systems, where author of the content had to define the complete domain description, user characteristics and adaptive behavior. This isolation led to very heterogeneous solutions and representations.

Different applications usually store heterogeneous user characteristics and preferences. Moreover, similar characteristics can be referenced by different names in multiple independent applications. To be able to merge the information we have to line up the user models.

In legacy adaptive systems this was impossible to achieve. A solution is offered by meta-data based approaches in open corpus adaptive hypermedia systems [26]. They make a use of semantic techniques and languages. Assuming that we represent the user models by ontologies, we can make use of the explicitly defined semantics. There are methods to merge ontologies and tools like ATOM [140], PROMPT [128] or SMART [127] have been developed. The overview of ontology mapping challenges and survey of the up-to-date tool is presented by Amrouch in [5].

2.9.2.3 Open information space

The ability to integrate with external public data is dependent on open world assumption. This means that we view things from another perspective than we are used to in closed environments like relational databases. In the open world, we assume that we never have a complete information. That means that a missing information does not lead to any conclusions. Allowing open world reasoning has interesting consequences, eg. the "Little House Problem" explained in [49]. In this case, under-specification in the knowledge base does not prevent the reasoner from obtaining the correct result. Not all reasoners are compliant with operational semantic of description logic and implement a closed world reasoning. To limit the open world to have a complete information on a node, the "negation as failure" operator has been defined [123].

For adaptive hypermedia, the difference usually means that while in the "closed world" we have a permanent, unambiguously defined domain, on the other hand "open world" is defined by extensible lightweight domain ontologies.

2.9.2.4 Annotation and reasoning

To add semantics, data is extended with meta-data. Meta-data can be used by software tools and agents in the reasoning process. It makes the information easier for computer processing. Meda-data is expressed by annotations. Annotations usually need to be created by humans and this is quite demanding task. To annotate large data, games are sometimes used [148]. Humans like playing games and recent research has utilized an entertainment to a useful activity. Annotation data is a very important prerequisite to successfully make use of semantic data in applications.

Annotations and labeling can be very useful to ensure a trust value of the resources in a distributed collaborative environment. Trust is the topmost layer of Semantic Web, and it is difficult to achieve. Researches in [38] have proposed an environment, where the trust is computed based on user labeling of resources and relations between the users modeled using FOAF (Friend of a Friend) ontology [20].

2.9.2.5 Collaboration and user groups

Usually, user models for individual users are maintained in adaptive applications. There has been a special type of the user model, the stereotype model, from the very beginning of adaptive system research era. In the stereotype model, there are users organised in groups with similar characteristics. However, to model the groups of users more dynamically, we need more sophisticated methods.

2.10 Summary

Adaptive hypermedia has been an actual research topic for over 15 years. During that time, various adaptation methods and techniques have been introduced. The research has been evolving and started to overlap with adjacent fields.

We have reviewed various application areas, where adaptive hypermedia can help users to improve the efficiency of their work and contribute to a better user experience. We have explained personalization methods and techniques in the scope of up-to-date classification and the important aspect of user adaptive systems – the user feedback.

In the last decade, number of reference models for adaptive hypermedia has been developed. We have explained the most important of them, compared the characteristics and discussed the shortcomings and possibilities for improvements.

The most important aspect of our work is bridging the adaptive hypermedia and semantic web research fields. We have explained the benefits gained by this fusion and identified issues of legacy adaptive hypermedia that can be solved within the new perspective. This is an important starting point for our research.

CHAPTER 3

The Generic Ontology-based Adaptive Web System Model

3.1 Adaptive Features and Behavior

The models mentioned in chapter 2 are (except GAM) unsuitable for general use, and all of them are missing specific important features. Therefore, an improved, generic reference model is needed to capture, both formally and informally, the important abstractions found in the existing and future Adaptive Hypermedia Systems. In this section, we will outline features that we consider necessary for adaptive systems and that constitute the main aspects of our novel Adaptive Hypermedia design.

3.1.1 Meta-model

Many existing adaptive hypermedia systems perform an adaptation of content and an adaptation of links. However, recent research noticed the need to select the type of adaptation itself. The systems should be able to vary the adaptation technique based on various factors.

Meta-model for adaptive hypermedia systems has been proposed by de Assis and Schwabe in [51]. Their model is based on the AHAM model and on the Munich Reference Model. However, these reference models were developed before many new aspects in Adaptive Hypermedia, which cannot be expressed in terms of these models, emerged.

The requirement to adjust the adaptive behavior of the system leads to a meta-adaptive system. Such a system extends the basic structure of adaptive systems thereby identifying their features with respect to the supported adaptations. This is the first step to a meta-adaptive system skeleton, which possesses at least the main components. However, in particular systems, additional extensions to the meta-model are necessary to accommodate specific features.

Meta-adaptation features require the system to be able to evaluate its own adaptive behavior. An important role in meta-adaptation holds user's feedback.

Example 3.1 (Meta-adaptation)**.** In an adaptive learning system, the meta-adaptation can be oriented on different learning styles of students. Various learning styles, e.g., auditory, visual, or tactile, may require adding different type of content, while performing the "inserting of fragments" adaptation.

Our architecture supports meta-adaptation natively through the abstraction of adaptive algorithms. We will show in Chapter 4, that by implementing algorithms satisfying the defined interface, any algorithm can be replaced based on the User Model state and rules definition. This results in the intended meta-adaptation.

3.1.2 Human memory aspects

In adaptive systems, various user characteristics are used in the adaptation process. Frequent property is the amount of user knowledge of some topic. This characteristic is mostly used in educational systems, thus we will focus only to these systems at first.

Example 3.2 (Memory aspects)**.** Adaptive learning system behavior can be improved by considering the principles of human memory. Typical user-adaptive system saves user's knowledge of a concept as a fixed value. The value is created when a user visits a page, or passes a test. In a programming course, students learn about data structures, and they are able to build a linked list at the end of the lesson. A student passes the test, let's say with an average result, and his knowledge value is stored to his user model. One month later, students should learn about advanced data structures such as the stack and the queue. Does the previously mentioned student still have the same knowledge about the linked list and is he able to use it to build the stack? He will probably need a short review of the previous topic to refresh the knowledge. This can become more obvious when considering longer time periods.

In the process of learning, the limited capacity of the human memory causes students to forget part of the knowledge acquired during the learning process. However, most current adaptive learning systems assume that the amount of user knowledge only grows. The process of repeating needs to be integrated into the system. Such extension was made by Bielikova and Nagy for the AHA! system [16].

In their work, they described the human mind using the Atkinson-Shiffrin model, which defines the following basic components: sensory memory, working memory and long-term memory. The amount of remembered information depends on time. The dependency has a characteristic of a falling exponential curve, which is called forgetting curve. Using this theorem, they could deduce a probability determining whether the information has been forgotten.

Our hypothesis is, that the fact of losing information is important not only in educational systems, but in adaptive systems in general. It does not have to represent only the process of forgetting some information. Losing information can be also observed during the changes of user's attitudes, gaining of new knowledge or after the change of some features offered by the system. Therefore, repeating and renewing of stored information needs to be integrated into the adaptation process.

Our model supports modeling the memory in two ways. Each User Model attribute has a timestamp, that can be used by the adaptation algorithm to assume the data validity or adjust the value depending on the feature characteristics and the time span. More advanced characteristics can be modeled as the time dimension. Within the dimensional model, history values can be stored, aggregated and used to infer implicit features, e.g., the student's learning progress.

Modeling of human memory is, however, an optional feature. First, the data can be organized in a temporal manner, or the history can be omitted. Second, the context-based meta-adaptation can be used to customize the perception of history data relevance. For example, a learning course should expect a student to forget knowledge not reused or repeated for a long time. Therefore, adaptation should head towards additional explanations or links that the student can follow to refresh the knowledge. On the other hand, a news portal requires a different interpretation of the user's memory model conception. Historical information can become outdated, and it is not required to refresh. At the same time, on the news portal, we assume links to related information, in order to guide the user in the information space. That means, linking a known information is not wrong. Any interpretation depends on the selection of adaptation algorithms for a specific use-case and is compliant with the proposed model.

3.1.3 Privacy and security issues

In the field of adaptive systems, we need to deal with the problem of ensuring user's *privacy* and data *security*. Within the user modeling process, data about the user is stored by the system and transferred over a network. The user provides person-related, possibly even very sensitive information about his disabilities and interests to the system. It is necessary to protect this information from unauthorized access. Security in user modeling is often seen as a precondition for realizing privacy [71].

As listed in [69], for the security of data transmission over a network various solutions exist, which are designated for different layers of the ISO/OSI reference model [50]. For our purposes, the privacy techniques performed by the application itself are more interesting.

An overview of up-to-date solutions for privacy of personalized hypermedia systems can be found in [161].

In [69], the author states the most important features, which the adaptive application should offer in terms of the privacy guarantee. Users should have the possibility to access the system anonymously. The user should be asked, whether user modeling for adaptation purposes should be performed. There are three possible levels: 1 – user modeling and adaptive behavior are disabled; 2 – short term user modeling (new model is created for each session); 3 – long term user modeling (user model is stored and reused in user's next session). At the end of each session the user should be asked whether he wants the user model to be stored for future use or deleted.

Different levels of anonymity can be required depending on the type of adaptive system. In [145], six different levels of anonymity are defined: 1 – *super-identification*, the user's identity is authenticated by a third-party system, 2 – *identification*, user identifies himself by a knowledge of secret, 3 – *latent identification (controlled pseudonyms)*, after the user identifies himself, he is assigned a system-defined pseudonym, 4 – *pseudonymous identification (uncontrolled pseudonyms)*, as part of the first access, the user decices on a unique pseudonym and secret, 5 – *anonymous identification*, user gains access by providing a secret, 6 – *anonymity*, the user neither identifies nor authenticates himself.

A user should be able to view and maintain the information stored in his/her user model. This will make the user characteristics kept within the system more transparent and allow an advanced user to understand the adaptation process better. The user needs to have the possibility to give explicit feedback to the system. Additionally, considering data integration, the user should be able to influence the visibility of his characteristics and personal information. This can significantly contribute to a better security of personal data. Sharing a User Model is a required and a very challenging task, where user's privacy cannot be violated. Fortunately, the users are willing to share personal information [106, 81]. It is important to find a balance between information revelation and personal privacy.

3.1.4 Individuals and groups

In adaptive systems, we need to model the users as individuals and even as groups with similar interests. The representation used in the user model needs to be relatively simple in order to be able to measure structural similarity between user models.

> **Example 3.3** (Group-based user modeling). Consider a learning application that is used continuously as a learning support tool for several terms. The system does not have much information about new students registered in the application. However, after a short-term analysis of the student's progress, he can be compared to previous terms' students with similar progress. This is an important step, providing the possibility of performing adaptation based on more complete characteristics of other users.

In case of navigational behavior in educational applications, the user characteristics can be represented as a transition matrix as proposed in [96]. This matrix can be visualized by a directed graph where the nodes are the pages, and the links are annotated by the number of traversals. For the purpose to find relevant relations in data, the Pathfinder Network Scaling (PFNet) procedures were developed.

Modeling of groups can be accomplished in a similar way to the modeling of individuals. The transitional matrices for all group members are added, and the PFNet procedure is applied to this new matrix.

3.1.5 Semantic model

The semantics is the study of meanings. Considering data in the context, the message behind the words can be understood. A semantic representation of data stored in the model is needed to allow the system to understand the data. The system should be able to process the data and exchange particular data fragments with other adaptive systems. The description using ontologies is more suitable for the meta-reasoning and allows direct manipulation [52].

Example 3.4 (Data semantics). We show the benefits of semantic data on an example of two independently developed applications. One of the applications is a learning system, second is a project assignment system. Without semantics, the administrator has to manually convert data, e.g., assign students' results to learning course lessons by topics. With the semantic web model, it is possible to base the data description on a common vocabulary, and use ontology to define contextual relationsips between both systems.

We propose a multidimensional User Model representation, where the user-specific data are defined by multiple lightweight ontologies. This architecture is beneficial for data integration, or rules definition, see Example 3.5.

The particular application may choose only part of the ontology, but the ontology as a whole has to be independent from the application. The semantic meaning will make the elements interchangeable and the possibility to interchange semantically annotated rules will enable to proof the validity of the reasoning.

3.1.6 Integration aspects

As the information about users is typically tightly connected to the domain, most of the adaptive hypermedia systems do not make the integration with other systems possible. To allow integration of multiple adaptive hypermedia systems we have to consider following characteristics. We have to separate the user model and the domain model. Usually, the presentation preferences can be separated from the domain and reused in other systems. However, the navigation preferences are dependent on the system architecture and cannot be reused easily. We have to make possible mapping between user data originating in different systems even if the information granularity differs.

Recently, motivated by the expansion of mobile and ubiquitous devices, the researchers noticed the need of sharing personal profiles of users, to enhance the adaptation abilities of user-adaptive applications. Various techniques of sharing personal data have been proposed [34, 151, 40]. The main obstacle is usually to solve the data interoperability problems. There are currently two main approaches. The first possibility is the way of standardization. The systems need to adhere to a fixed representation that needs to be respected by all service providers. The second approach is using mediation techniques to transfer the data from one representation to another.

Standardization-based approach of AHS integration assumes a common semantic representation of user models within all participating systems, usually expressed by a shared ontology [89]. Implementing domain models of adaptive systems as ontologies is the first step toward interoperability. A standardized user-modeling ontology is a possible solution to make the information exchange possible. However, the fundamental requirement of this approach is that all participants agree upon the standardized ontology, which may pose an issue for some of them [13].

As a generalization of a standardization-based approach, the central user modeling server can be assumed. A solution, where adaptive systems do not need to support user modeling was used in [100]. The networked adaptive applications act as clients, they simultaneously update the central user model on the server, and they can request back personal information when needed. Another solution based on a central server is presented in [157], where the exchange of user data between applications is supported by Semantic Web technologies. The authors call the component providing user model storage the Generic User Modeling Component.

More complex solutions than standardization are utilized by mediation [14]. Without any standard vocabulary, it is necessary to solve syntactic and semantic heterogeneity issues. Ontology mediation is the process of reconciling differences between heterogeneous ontologies in order to achieve inter-operation between data sources annotated with and applications using these ontologies [56]. To overcome the heterogeneity of user modeling data, two steps are required. First, the reasoning and inference mechanisms for converting data between various representations, applications and domains need to be developed and applied. Second, the semantically enhanced knowledge bases are exploit, facilitating the above reasoning and inference [13].

Although, there have been projects like [89] primarily focusing on the standard, widely accepted user modeling ontology, [114] claims that the standardization approach is not a feasible solution. Such statement suggests to follow the second direction, the mediation of different domains based on natural language processing and artificial intelligence. However, there is also a possibility of a hybrid approach combining both standardization and mediation approaches. Such unification is the aim of our approach.

3.2 The extended modeling loop

In the previous section, we described the most desirable requirements of good adaptive systems, which are missing in most of the current solutions. In this section, we will describe our model proposal, which aims to be a generic fundamental for adaptive hypermedia systems and embodies all the requirements.

In our research, we based our general model on the GAM [61] foundations. We intend to make it a powerful tool for designing adaptive systems, by extending this model with new functions, new approaches and Semantic Web technologies. GAM is a state-machine based model, which, compared to other models, is more low-level. We based our work on GAM, because GAM is formally described, provides an explicit interface model and the concepts of push and pull adaptation. The model assumes systems to be interactive (systems where a user interacts with the system).

In the description of the GAM model in Chapter 2, the evolvement of the modeling loop was outlined. Our proposal follows and improves the works of Brusilovsky [22] and de Vrieze [61]. In Figure 3.1, our additional extensions to the existing modeling loop can be seen. Brusilovsky in his loop presented a system, that collects and processes user data, and he divided the processing part to User Modeling and Adaptation. De Vrieze identified, that the User Modeling process needs to be separated from the whole system. The system knows, how to change itself according to the user. However, the User Modeling component knows how to answer questions about user. We have further extended the User Modeling process to highlight important features of Adaptive Hypermedia Systems. The detailed description of our extensions in the modeling loop follows.

The core of our architecture is represented by an overlay user model and rules. Our system design is generic and can utilize various data representations. However, to utilize the most benefits of the proposed architecture, as mentioned in Section 3.1.5, the user data and rules can be represented by multiple lightweight ontologies organized in orthogonal planes of a multidimensional matrix (1). The stored information is used in the adaptation process and also for choosing the right adaptation technique (2). This allows meta-adaptation, see Section 3.1.1, and an appropriate algorithm for modeling a specific user can be chosen. Additionally, putting the system in a specific context can help to infer more accurate information (3). Some of the necessary information can be calculated on demand from the stored data. This process can be viewed as answering questions about the user (4). Subsequent advantage of our architecture is the possibility of using other server's information (5). Since all the data can be semantically annotated, it adds the possibility to enhance the User Modeling process with User Model information offered by a remote system. For integration methods see Section 3.1.6. The distributed information should contain not only the user data, but also semantically annotated rules describing the reasoning process that led the remote system to the user data values (6). Sometimes, the result obtained by a heuristic algorithm can be false. Therefore, it is a good practice to receive feedback on the performed changes from the user (7). Our architecture supports effortless extension of the adaptation component with custom algorithms. For example, it would be helpful to add uncertainty measuring algorithms into the component (8).

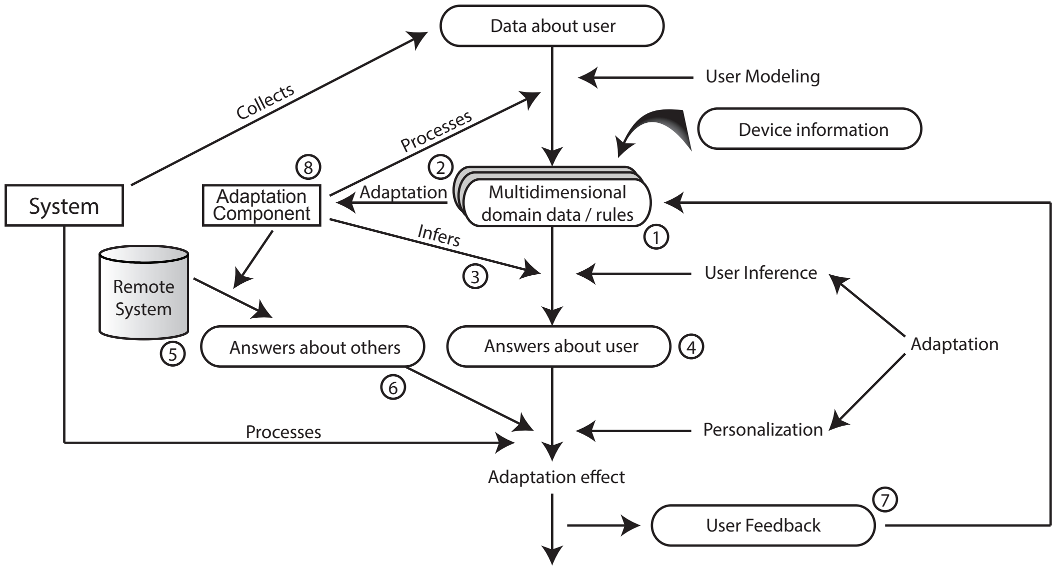


Figure 3.1: Our extended modeling loop.

The proposed model is based on the semantic data representation. We want to achieve the ability to interchange the data between systems and reuse the content. Therefore, we propose to extend the user model with an ontology, which can be understood by various systems.

3.3 Adaptive System Model

This section presents a detailed description of the Generic Ontology-based Model for Adaptive Web Environments (GOMAWE[1]). Important characteristics differentiate the model from other works in the Adaptive Hypermedia research area. First, the model

[1]Gomawe is a New Caledonian god who created humans. Similarly, our model will be used to create adaptive web systems for humans.

is generic, which means it is modular, extendable and applicable to support all up-to-date adaptation techniques and applications. Second, the model is ontology-based, which is an important feature to make integration of AH systems and exchange of user-specific data possible. Finally, the model is based on a formal specification, and its application is supported by Adaptive System Framework that will be described in detail in the next chapter.

Architecture of the proposed model can be divided into several layers as depicted in Figure 3.2. The first proposal of the GOMAWE model was presented in [A.5] and a revised formal specification in [A.11]. The model is logic-based and can utilize semantic reasoning in the user modeling process. The abstraction of the model and the object-oriented design support component-based implementation alternatives.

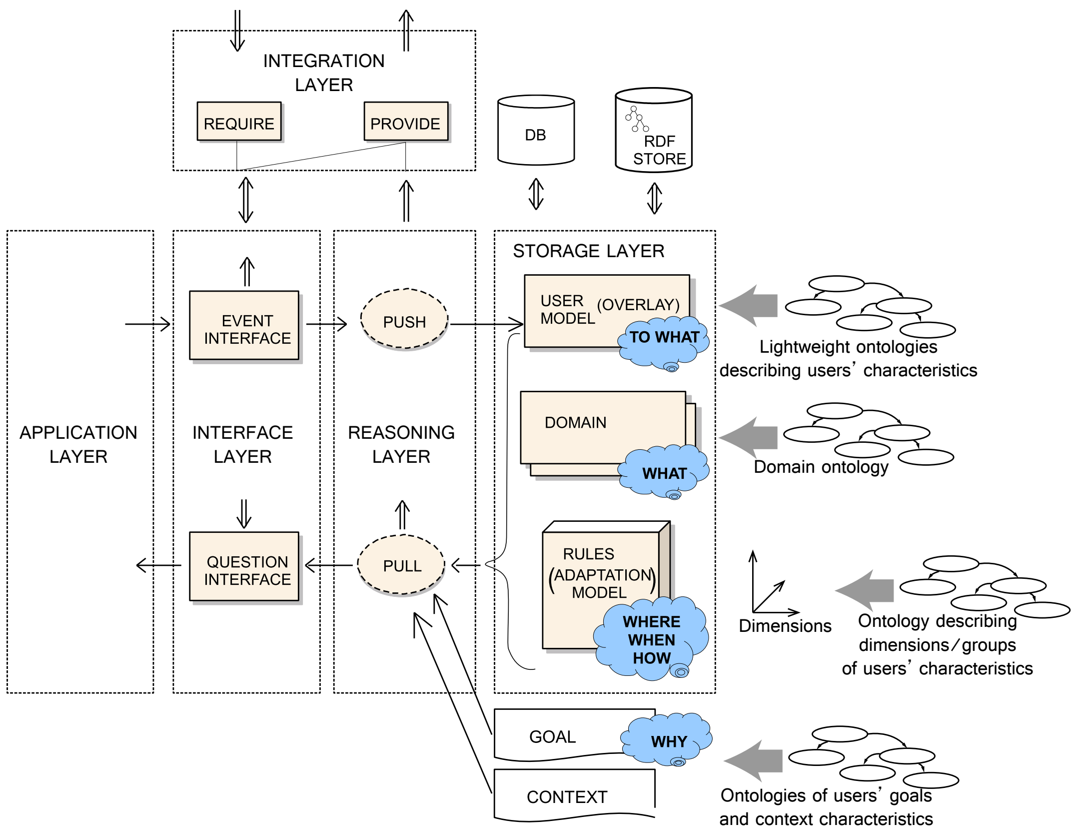


Figure 3.2: Overview of GOMAWE.

Following the process model described by Knutov [103] and the research questions formulated in sections 2.2 and 2.3, we have annotated components of the model to point out the related questions. This explains which element of the model is responsible for a specific concern, see Figure 3.2.

3.3.1 Semantic user model

The GOMAWE is a model based on the semantic data representation. Such representation can be utilized by machines in process automation, data integration and reuse of knowledge across applications. The machine understandable contents are called metadata, and their semantics can be specified using ontologies. Ontologies play an important role in the Semantic Web as they provide a common conceptual model of a domain that can be shared by applications. Ontology can be used to reason about objects in the domain and the relations between them.

The important part of our model extension is the storage layer. The data structure is represented by an ontology, which enables the storage of metadata together with the data. Furthermore, the ontology is not a monolithic detailed ontology, but the data structure consists of multiple lighter-weight ontologies, which can be used together. These ontologies should be independent, modular and layered. Selected ontologies appear as layers on the dimensions of a multidimensional matrix described in the next section. The user model is, in fact, an overlay model consisting of instances of objects described by the ontologies. The semantic annotation of data and rules is an important extension of the model. It enables the definition of global terms typically used among the adaptive systems.

The user model keeps the user characteristics. To split the management of user properties into a more granular structure, we distinguish user profile and user model as independent components. The user profile is used to store the preferences set explicitly by the user. This corresponds with the adaptable behavior of the system. On the other hand, the user model is used to store data originating from the automatic user monitoring process. We take this as the adaptive behavior of the system. Implicit and explicit user feedback is part of the process.

Adaptation techniques can be based on rules that define the adaptive behaviour. The rules are assumed to be persisted in the storage layer. Data represent the conditions and possible adaptation effects. Additionaly, rules can be organized in a multidimensional hierarchy that will be explained in the next section, and semantics can be provided by ontologies. The actual adaptation is perfomed within the reasoning layer by algorithms that use user-specific data and rules.

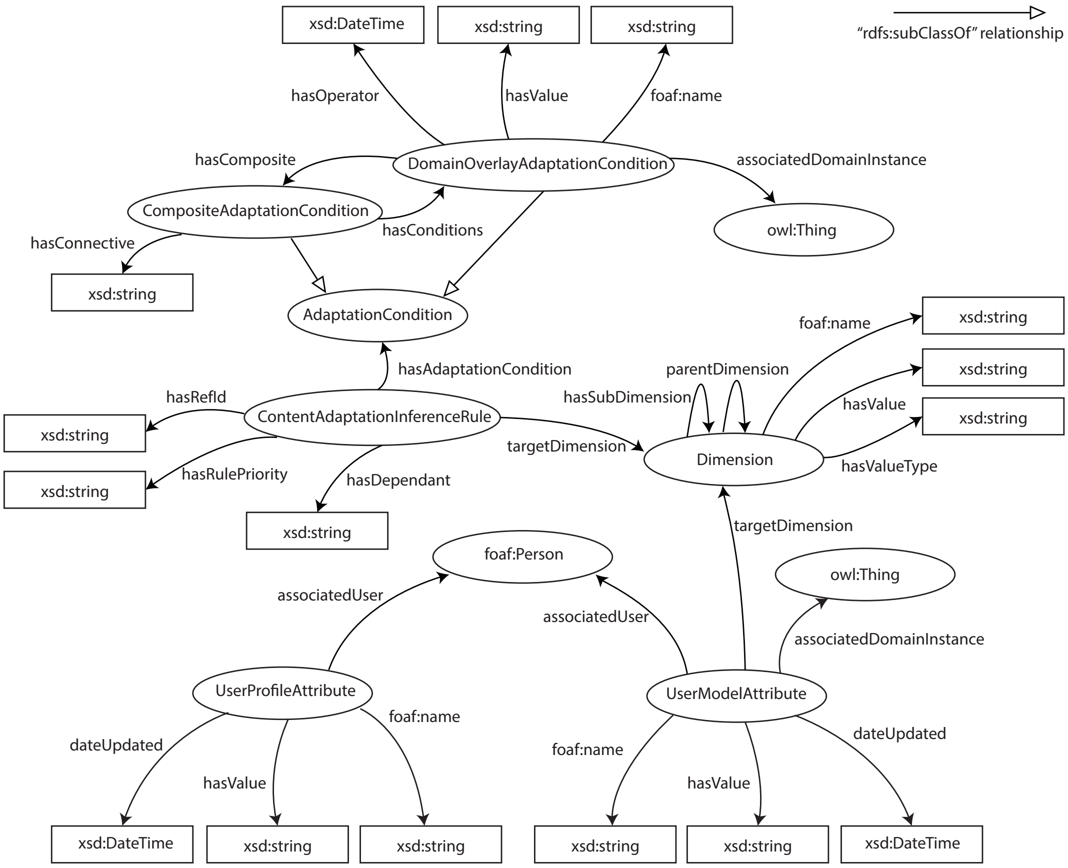


Figure 3.3: Data storage ontology – User Model, User Profile and Inference Rules.

We have created an ontology describing the basic entities of the User Model and Rules, see Fig 3.3. This ontology describes the system architecture and has been proved useful for data exchange, see Section 5.4.2. Other ontologies shown in Fig. 3.2 are application dependent. They can be used to describe the domain, user characteristics, context properties, rules, etc. Some of them can be defined by the system developer, however, a recommended best-practice is to reuse existing, widely accepted ontologies.

3.3.2 Multidimensional hierarchy

In our reference model, the User Model can be described by multiple lightweight ontologies. Thanks to the modular structure, only a sub-part of the most generic ontology needs to be selected. Objects of these ontologies are instantiated depending on actual needs. Selected ontologies appear as layers on the dimensions of a multidimensional matrix. This structure is then used to select corresponding rules and thereby infer further information not included in the User Model. The design was inspired by the semantic framework proposed by Carmagnola in [36].

All the information is stored in a hierarchical way, i.e., at the first level, there is usually the definition of general concepts and on the following levels, there are specialized concepts defined. This will enable for the particular application to instantiate only the related concepts as a subset of a general unit.

To make further inferences and to express further relations not provided by the ontology, the reference model defines two types of rules that are used to define the adaptive behavior. The first type is the Inference Rule. The Inference Rule is referred by a unique identifier and based on a condition the rule defines an alternative content. The second type of rule, the Event-Condition-Action Rule is a rule triggered by an action. For detailed description and formal definitions of rules, see Section 3.3.3.3.

By virtue of the layered structure of the model, it is also possible to store various adaptation types in a designated plane and support meta-adaptation in that way, see (2) in Fig. 3.1. The concept of meta-adaptation was described in detail in Section 3.1.1. In the first phase, the adaptation component performs a query on the specific part of the ontology and selects suitable adaptation techniques, which will be used in later phase. Moreover, some information about device capabilities and properties can be stored in that way, see (1) in Fig. 3.1.

The basic idea of the matrix is that the rules can be defined on the points of intersection between planes. The rule specifies classes or properties of classes that contribute to define the value of the inferred feature. Each plane can represent the particular type of knowledge, e.g., information about the user, his characteristics, preferences, or goals, device information, etc.

Example 3.5 (Inference Rule)**.** To demonstrate the multidimensional hierarchy, let's explain the exemplary situation depicted in Fig. 3.4. Let's have an adaptive e-learning system. We will consider three ontologies, to be able to display the situation graphically in three-dimensional space. This, however, does not limit the number of dimensions for other situations. In our example, we want to customize the content based on selected user's characteristics. We will consider the following ontologies: *user characteristics, environment description* and *user's learning style.*

On the X1 plane, there is the ontology of the user characteristics (gender, age, occupation), on the X2 plane, the ontology of the environment description (time, location), on the X3 plane, the ontology of the user's learning styles. Using this structure, we can provide a rule that defines the dependant - the customized hypermedia element for the specific user. In our example, the rule condition is defined as: *(X1-plane UserCharacteristics <occupation=student>) AND (X2-plane Environment <datetime=5.5.2008,9:00, location=classroom>) AND (X3-plane LearningStyle<cognition=auditory, communication=interpersonal>)) Then (dependant <URI>)*, where URI is the reference to an alternative content suitable for the given characteristics, see Fig. 3.4.

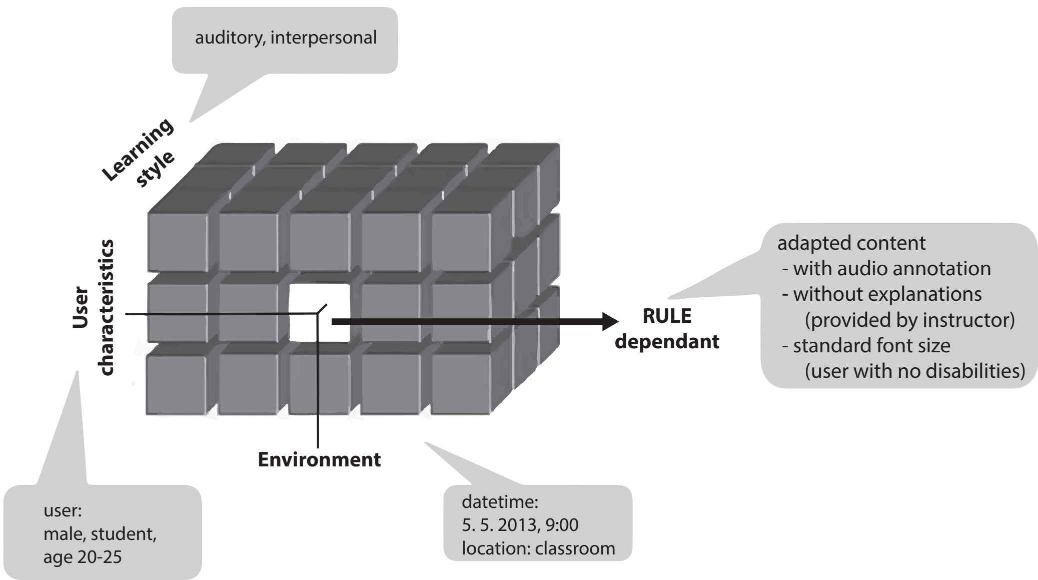


Figure 3.4: Multidimensional matrix example (three dimensions).

Example 3.6 (Event-Condition-Action Rule)**.** The second type of rule can be used to infer new User Model attribute values and to update the current User Model. As an explanatory example, we will consider a similar situation to the previous example in an adaptive e-learning system. In our example, we want to know, for the purpose of content adaptation, whether the user knows basics of SQL query language. The Event-Condition-Action rule consists of three parts – the action is the initial event leading to evaluation of the rule, the condition is the necessary consideration that needs to be satisfied before the action is executed, the action defines the possible new information that can be used in the adaptation process or that can be stored in the User Model. In our example, the triggering action can be the user's login or entering a lesson page. Based on the condition that includes environment description, e.g., location and date, and user's characteristics, e.g., he is a student, a new user model attribute about the knowledge is inferred. The rule can be written as login $\xrightarrow{student, date=5.5.2013, location=classroom}$ knows SQL.

3.3.3 Components specification

Elementary components identified in an adaptive system are domain model (DM), user model (UM), content unit model (CuM), adaptation model (AM) and presentation model (PM).

The GOMAWE can be divided into five layers, see Fig. 3.2. In the following subsections, we will define sub-models of the storage layer.

3.3.3.1 Domain model

The Domain Model in the proposed architecture defines the conceptual framework and semantics of hypermedia content. To express the semantics, domain ontologies can be used. To be able to perform specific content-adaptation techniques, e.g., altering, inserting or removing fragments, fine grained portions of the content are needed. We call these fragments *hypermedia elements*. The elements are identified by URI, can be distributed over network and reused in a particular content. The elements are referred to by atomic domain concepts, and these atomic concepts can be grouped to composite concepts.

Definition 3.1 (Concept)**.** A Concept is a tuple

$$c = (P, C_{SUB}), \tag{3.1}$$

where $c \in C$, P is a set of concept properties, and C_{SUB} is a set of subconcepts. If $C_{SUB} \neq \varnothing$ the concept is called *composite concept*. If $C_{SUB} = \varnothing$ the concept is called *atomic concept*.

Definition 3.2 (Concept Property)**.** A Concept Property is a tuple

$$p = (a, v), \tag{3.2}$$

where $p \in P$, $a \in A$, where A is a set of class attributes defined in the domain ontology, $v \in V_a$, where V_a is the range of attribute a, and $V = \bigcup_{a \in A} V_a$ is a set of all attribute values.

Definition 3.3 (Set of Required Properties)**.** The Set of Required Properties $P_m \supseteq P$ is a set of mandatory concept properties required to have a non-empty value.

Definition 3.4 (Hypermedia Element)**.** The Hypermedia Element e is any resource, i.e., document, accessible over network and identified by an unique URI.

Definition 3.5 (Default Hypermedia Element)**.** The Default Hypermedia Element e_d is defined by a function

$$e_d : C_A \to R | \forall c \in C_A : e_{d(c)} \in R, \tag{3.3}$$

where C_A is a set of atomic concepts, and R is a set of resources.

Definition 3.6 (Domain Model)**.** The Domain Model is a tuple

$$DM = (C, D, R_C, R_D), \tag{3.4}$$

where $C = C_C \cup C_A$ is a set of concepts (composite and atomic), D is a set of domain concept instances, R_C is a set of relations between the concepts, R_D is a set of relations between concept instances.

Concepts are unary predicates denoting entities or classes. They describe a set of properties. Relations are binary predicates expressing relations between concepts or their instances. Concept relations include inheritance, containment and reference relationships. An example of concept relation is composition, e.g., lesson contains chapters. An example of instance relation is a prerequisite, e.g., "Chapter 1" should be learned before "Chapter 2".

Our view on concepts is from the object-oriented perspective. As the GOMAWE model is generic, the actual implementation and representation of domain model data can significantly differ. With limited system features, the data can be stored in a relational database, or it can be stored in a triple store, where representation using RDF, or OWL can be used. Evaluation has been performed for relational database and RDF triple store, see Chapter 5. The domain model is assumed to be final and fully defined. Therefore, e.g., in the case of RDF, the actual application-specific domain model is a subset of the RDF graph. The RDF graph can contain additional information not mapped to the entity classes. This limitation can be solved by more sophisticated mapping approaches. One possible solution is using the OWL2 constraints, as presented in [110]. Such extension is still compatible with our model definition.

3.3.3.2 User model

In our design, the user model has a special architecture that was devised from the requirements, including types of stored information and methods of information retrieval. We divided the user data into two parts – the User Profile and User Model. This division corresponds to explicit and implicit personalization styles. Implicit personalization is performed by the adaptive system. On the other hand, explicit personalization is performed by the user using special features of the system. A system with such personalization features is called adaptable system. Our architecture can be perceived as a hybrid solution combining both personalization methods.

The User Profile contains explicit user's preferences, i.e., preferences explicitly filled by the users. This data is stored as key-value pairs, see Definition 3.7. The key is usually a constant string defined by the developer of the application. Typical origin of the data is the "settings page" of the application. However, when we assume integration with other personal data providers, the User Profile data can be filled utilizing social networks' user profiles available in services like Facebook, Twitter or LinkedIn.

Definition 3.7 (User Profile)**.** The User Profile of a user u is a tuple $UP_{(u)} = (A, V, r)$

$$r : A \to V | \forall a \in A : r_{(a)} \in V_a, \tag{3.5}$$

where A is a set of attributes defined as the vocabulary of user's characteristics, $V = \bigcup_{a \in A} V_a$ is a set of attribute values and V_a is the range of attribute a.

The User Model denotes in our terminology a model containing implicit user's characteristics. While the explicit characteristics are set by the user oneself, the implicit characteristics are devised by the adaptive system. The system collects various values related to a specific object of the domain. The Definition 3.8 denotes a simplified flattened User Model.

Definition 3.8 (User Model)**.** The User Model of a user u is a tuple $UM_{(u)} = (D, A, V, r)$

$$r : D \times A \to V | \forall a \in D \times A : r_{(a)} \in V_a, \tag{3.6}$$

where D is a set of domain concept instances, see Definition 3.6, A is a set of attributes defined as the vocabulary of user's implicit characteristics, $V = \bigcup_{a \in A} V_a$ is a set of attribute values and V_a is the range of attribute a.

The user model is an overlay model. It covers part of the domain and defines characteristics of the user. Examples of information stored in a user model are the degree of knowledge of the user about items in the domain, user goal, interests of the user or his preferences.

An extended version that is used in our architecture follows the principle of Multidimensional Matrix [A.5]. The characteristics are categorized to a set of dimensions, see Definition 3.9. The assignment of attributes and dimensions is utilized in the user modeling process.

Definition 3.9 (Multidimensional User Model)**.** The Multidimensional User Model of a user u is a tuple $MUM = (X, D, A, V, r, s)$

$$\begin{aligned} r &: D \times A \rightarrow V | \forall a \in D \times A : r_{(a)} \in V_a, \\ s &: X \rightarrow A | \forall x \in X : s_{(x)} \in A_x, \end{aligned} \tag{3.7}$$

where X is a set of dimensions, D is a set of domain instances, $A = \bigcup_{x \in X} A_x$ is a set of attributes defined as the vocabulary of user's implicit characteristics, each associated with a particular dimension, A_x is the attribute range of dimension x, $V = \bigcup_{a \in A} V_a$ is a set of attribute values and V_a is the range of attribute a.

Definition 3.10 (Content unit Model)**.** The Content unit Model CuM is a complement to the user model related not to the user, but to the individual items of the domain, e.g., ratings of items.

3.3.3.3 Rule repository

An improvement of the storage layer in our model is a multidimensional matrix storing adaptation rules. These rules can be used to deduce information not directly stored in the user model. Selected ontologies appear as layers on the dimensions of the Multidimensional Matrix. This structure is then used to select corresponding rules and thereby infer further information.

Definition 3.11 (Adaptation Model)**.** The Adaptation Model is a tuple $AM = (IR, ECA, AF)$, where IR is a set of inference rules, ECA is a set of Event-Condition-Action rules, and AF is a set of adaptation functions.

We can define an elementary adaptation as an adaptation function.

Definition 3.12 (Adaptation Function)**.** An Adaptation Function f_a is a transformation between default and adapted hypermedia elements. A hypermedia element, see Definition 3.5, is considered as a portion of HTML code that is a part of the web page.

$$f_a : e_d \rightarrow e_a, \tag{3.8}$$

where $f_a \in AF$, $e_d \in R$ is the default element, $e_a \in R$ is the adapted element, and R is a set of resources.

The Inference Rule is based on conditions. Conditions can be joined by logical operators, i.e., AND or OR. Based on the result of condition evaluation, dependant is returned if condition is satisfied, or the value is empty in another case. In our design, dependants are the alternative hypermedia elements that can be used in the adaptation algorithm, e.g., as altering fragments. Inference rules have a unique reference id, which can be assigned to a group of rules. This id can be referred from the content description, to apply adaptation on the particular concept.

Definition 3.13 (Inference Rule)**.** The Inference Rule is a tuple $IR = (id, C, D, X)$, where id is an unique reference id of a set of inference rules, C is a set of conditions, $D \supseteq R$ is a set of dependants, and X is a set of dimensions.

The Event-Condition-Action rule (ECA) is formed by an event, a condition and an action to be performed. This corresponds with "Active rules" – rules with a specific form: Event-Condition-Action. With utilization of active rules, the system is capable of integration of all existing techniques. Furthermore, this system allows introducing new adaptation strategies easily.

Definition 3.14 (Event-Condition-Action Rule)**.** The Event-Condition-Action (ECA) Rule is a tuple $ECA = (id, E, C, A, X)$, where id is an unique reference id of a set of ECA rules, E is a set of events that trigger execution of the rule, C is a set of conditions that need to be satisfied, A is a set of actions to be performed, and X is a set of dimensions.

This section formally defined the storage layer of GOMAWE. Next, the presentation layer is defined.

Definition 3.15 (Presentation Model)**.** Presentation model (PM) is a set of visual hierarchy (VH) elements. Elements are instances of domain model concepts.

$$PM = (VH | V_e \in VH : e \in C) \tag{3.9}$$

Visual hierarchy (VH) can be expressed using the Adaptive Hypermedia Document Template (AHDT).

$$\begin{aligned} Component &= (ContentUnit \mid Hypermedia\ element) \\ Container &= (Container \mid Component) \\ AHDT &= (Container \mid Component) \end{aligned} \tag{3.10}$$

The document template consists of containers and components. Containers are used to group components and to place them in visual layouts. Component is either a domain-related content unit or a hypermedia element, see Definition 3.4.

3.4 Evolution of AHS Models

The AHAM model [54] has been the most influential model in area of adaptive hypermedia. Later there have been some attempts to extend this model from the perspective of UML modeling [107], user's context [9] or user's interaction with the system [58].

Our improved adaptive hypermedia model GOMAWE can be compared to AHAM. If we view the GOMAWE model from a different perspective, we can align the layers with the layers of AHAM model and outline the enhancements, see Fig. 3.5. The three layers of AHAM are extended as follows.

The Application Layer corresponds with the AHAM's Run-time Layer and represents the specific workings of each particular application. The user-independent segment of the application is linked with the user-specific segment through the Interface Layer that provides an abstraction of the adaptation engine. The Integration Layer is added to support user-specific data exchange among multiple adaptive applications. The Reasoning Layer acts as an information manager. This layer is capable of providing data from and into the Integration Layer, just as from the Storage Layer. This layer consists of a range of adaptation algorithms and is easily extendable. Storage layer has similar sub-models as AHAM has – the Domain Model, the User Model and the Rule Repository, which can be compared to the AHAM's Adaptation Model. The important difference is, that each of the sub-models is described by ontology. Even light-weight

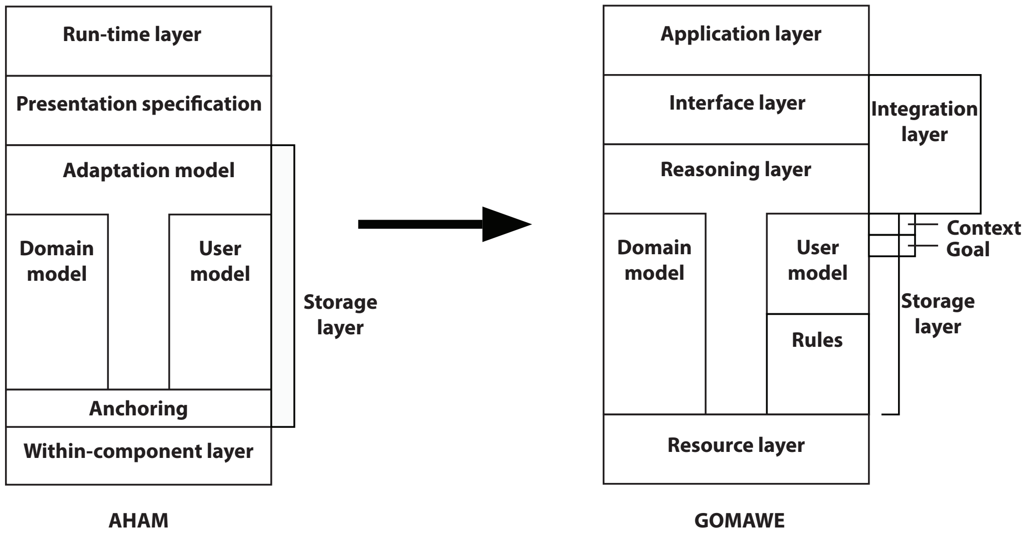


Figure 3.5: From AHAM to GOMAWE.

ontologies or taxonomies, can adhere to comprehension of concepts stored in the data model. The resource layer comprises of content resources. It can be interpreted as a fixed set of internal content resources, but it can also refer to external resources in a form of Web URL link.

3.5 Summary

This chapter described the original idea of this thesis - a novel adaptive hypermedia model GOMAWE (General Ontology-based Model for Adaptive Web Environments). This model was designed as a reaction to the design issues of classical AHS models. In the beginning of our research, we have identified important, but often overseen features of adaptive hypermedia. Inspired by the recent research advances, we have extended the schema of an adaptation loop, describing the adaptation process within the system. Next, we have designed an architecture of an adaptive hypermedia system - the GOMAWE model. All necessary features were incorporated into the design. The model was formally described. In following chapters we will provide information about extensibility, implementation and user-centered layered evaluation.

CHAPTER 4

The GOMAWE Extensions and the Adaptive System Framework

Adaptive hypermedia systems (AHS) are complex systems that require an expensive and time-consuming design and development process. Complex solutions are usually realized as reusable frameworks and program libraries. However, there is currently no widely acceptable solution for building AHS. Based on our research, we have developed a framework that could significantly make the design and development of AHS easier. First, we formalized the adaptive system architecture, and then, we defined basic structures for storing required data. Further, we designed the adaptation and integration modules and developed reusable adaptive web user interface components. Such a framework is considered to become a foundation stone for various types of AHS.

The GOMAWE model forms a theoretical basis for an application framework that should rapidly simplify and speed up the AHS development. This is achieved by reusable ready-to-use software components provided by the framework implementation and extensibility of the framework for further use cases and novel technologies.

The Adaptive System Framework (ASF) was built to help a software developer create adaptive web applications. ASF provides the most typical AHS components serving as building blocks for further development. ASF is based on the theoretical model and satisfies the following important requirements. To be generally applicable, the framework has to be split into components with independent responsibilities. To follow generally accepted solutions to common application problems, design patterns [72] should be extensively used. The implementation of the framework should be based on well-known and widely used application frameworks. In contrast to other frameworks focusing on the users' collaboration [160] or adaptation process modeling [104], our framework aims at formalization of adaptive system architecture. Further, it focuses on targeting the problem of a storage layer abstraction, foundations of the data structures needed for a user modeling and providing a basic set of adaptation-oriented user interface components.

In comparison with other solutions, our ASF project aims at supporting not only adaptive learning, but also adaptive hypermedia systems in general. The purpose of our project is to provide a reusable solution that could be used by the developers of adaptive applications.

4.1 Framework architecture

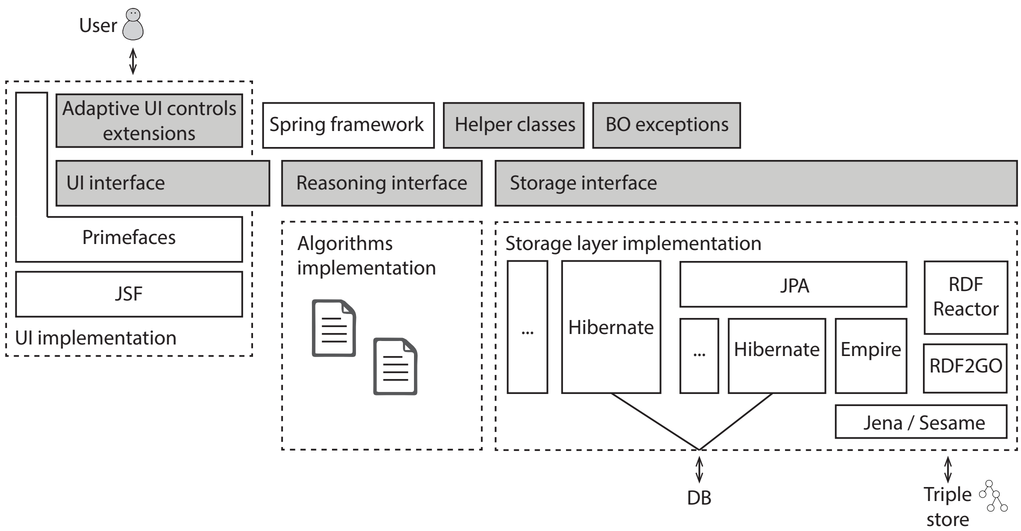


Figure 4.1: Adaptive system framework architecture.

The ASF architecture is generic. It defines an interface of various layers of the potential system, and therefore, its implementation can be realized in multiple programming languages. Fig. 4.1 represents the description of ASF architecture based on our GOMAWE model. The ASF architecture consists of the layers replacing the GOMAWE

Storage, Reasoning and UI interface layers. The main highlighted components of the core framework library define the fundamentals of the architecture. The default implementation is built on the selected persistence frameworks supporting relational database and triple stores. In our experiments, the storage layer was based on various frameworks, e.g., JPA, Hibernate, Empire, or RDF Reactor. However, our framework can be extended by any other implementation of the storage interface. The extensions are indicated by the '. . . ' symbol in the diagram; see Fig. 4.1. The same situation is in case of algorithms performing the adaptation above the storage layer. Some algorithms are part of the framework, others can be added by the developer as an implementation of the reasoning interface.

The user interface is based on Java Server Faces (JSF) and is supported by a Primefaces components suite[1]. Our goal is to extend basic web components by the adaptation-specific extensions; see Fig. 4.1. We think that there are several adaptation techniques that the framework can provide "out of the box" allowing the developers to apply the adaptation without any need of additional work.

4.1.1 Data storage

One of the most important parts of every adaptive system is the user model. This is the repository, where information about the user is stored. This information is used in the adaptation process to filter information and personalize the presentation according to the user's preferences.

We divided the user data into two parts – the user profile and user model. The user profile contains explicit user's preferences. This data is corresponding to the "settings page" and is stored as key-value pairs. The key is usually a constant string defined by the developer of the application. The user model, on the other hand, stores the data observed while the application monitors the user. The information is always associated with a domain object and represents the user's relation to the object, e.g., user's knowledge of the topic, user's preferences, or their past experience. The user model corresponds to the overlay over the application domain model.

The access to the user model and user's profile is supported by an adaptation manager. The adaptation manager implementation is based both on the Singleton Design Pattern providing a manager instance and the Factory Method Design Pattern used for creating domain-specific model instances for an individual user.

[1] http://primefaces.org/

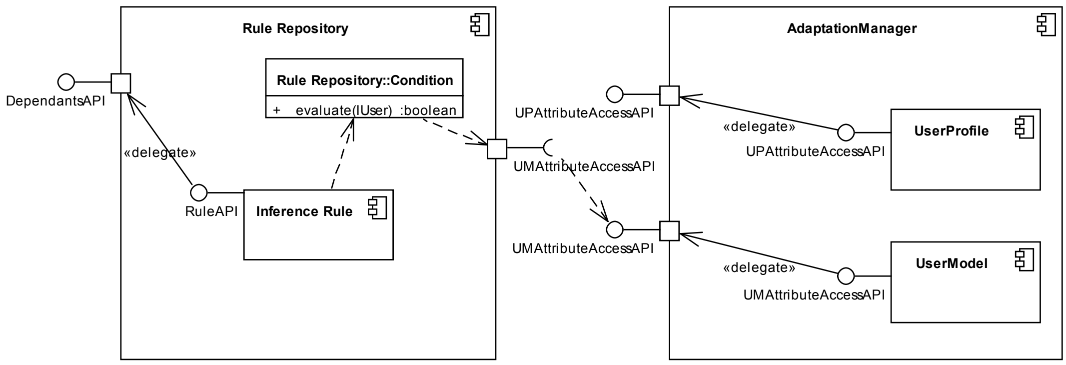


Figure 4.2: Components in the ASF storage layer.

In our design, the user model has a special architecture. Each attribute is assigned to one dimension. Dimensions can be custom-defined for each adaptive system. The dimension forms a group of related attributes. It can be visualized as a multidimensional matrix. The multidimensional user model was formally described in the previous chapter (Definition 3.9).

Another important part of the adaptive system storage is the rule repository. The rules can represent conditions defined by the author of the content of the application, e.g., by a teacher who prepares an adaptive course, or they can be generated by specialized adaptation algorithms. The rules assume that the user-model characteristics are associated with predefined dimensions. The dimensions can be used to filter the rules while the rules are being evaluated. This contributes to better performance and helps the designer of an adaptive algorithm to maintain the rules easier.

In Fig. 4.2, the main components of the Storage Layer are presented as a component diagram. Adaptation Manager is a component providing access to the user-specific data storage. It encapsulates both User Profile, see Definition 3.7, and User Model, see Definition 3.8. The manager abstracts the interface to avoid direct manipulation with attribute objects. The Rule Repository encapsulates all rules and provides an interface to access them. The Inference Rule, see Definition 3.13, depends on Condition Evaluation. To be able to evaluate the condition, interface to access User Model attributes of a specific user is needed. The Event-Condition-Action rules, see Definition 3.14, are not displayed in the figure. However, they depend similarly on the condition evaluation. These rules just provide another interface to the programmer, and their execution depends on actions rather than on direct reference as in the case of Inference rules.

Both the domain and the user model can be represented by simple concepts and their relations. However, our solution was designed to use multiple lightweight ontologies. The use of ontologies was motivated by the requirements of the data semantics, data exchange and integration among applications, and as well, by the need to infer the information implicitly stored in the user model.

The adaptive process is executed above the storage layer and acts as a mediator between the raw data and the user.

4.1.2 Adaptive behavior

One of the goals of the framework optimization is to make components of an adaptive system reusable and generally applicable. To achieve this requirement, we defined a general interface over any algorithm that will be used to perform the adaptation (Fig. 4.3).

The adaptation algorithms are further divided according to the "adaptation techniques taxonomy" (Fig. 2.1). A simplified version of the taxonomy is based on the taxonomy defined in [102]. Less important techniques currently not implemented by the framework were excluded from the original taxonomy. A content adaptation, link adaptation and link-group adaptation algorithms are specified in the framework.

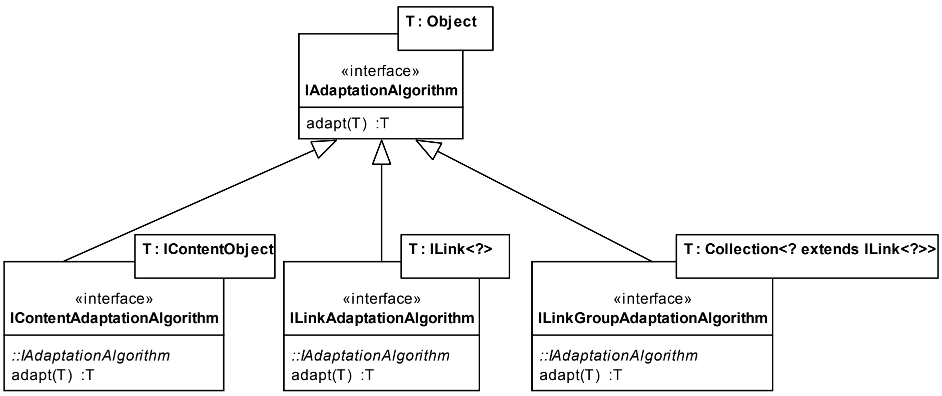


Figure 4.3: Adaptation algorithms classification.

A content adaptation algorithm is an algorithm which is used to transparently transforming the content of the domain concepts based on the user model. It can be used to substitute the elements of the domain concepts. For example, we can recognize various extents, based on the user's knowledge stereotypes (beginner, advanced, expert).

A link adaptation algorithm is intended for customizing a single link. On the other hand, the link-group adaptation algorithm assumes a collection of the links to serve as both input and result. The links can be adapted by sorting. The input collection is processed, and the order of the links is modified. A different approach is used for a link generation. In this case, the result is not dependent on the adaptation function input, since the data is retrieved from a repository. Other link adaptation strategies include a direct guidance, link annotation or adaptive link hiding.

There are two main types of the link adaptation algorithms in the framework:

- adaptation based on the current context, where the existing links can be sorted, filtered, etc.

- link generation, where the algorithm is responsible only for retrieving the input data. A default value can be provided to the adaptation function. It can be used in case when the user disables the adaptation, or if there is not sufficient input data to generate the links automatically.

In GOMAWE, the adaptation was designed as an extendable set of black box components that perform the adaptation based on the information stored in the user model. A framework implementation (Fig. 4.4) is realized as a Strategy Design Pattern [72]. The intent of the Strategy Design Pattern is, first, to define a family of algorithms, second, encapsulate each of them, and third, make them interchangeable. It enables the algorithm to vary independently of the clients that use it.

The adaptive process of the framework follows the formal specification of GOMAWE. The adaptation function (Definition 3.12) has been defined in the previous chapter. A generic group adaptation function usually takes a collection of default or initial values as an input, and returns an adapted collection of items of the same type. The particular algorithm is encapsulated inside the black box which is implemented as a class. Algorithm instances can be optionally parameterized before the actual adaptation is performed.

Another extension of the adaptation component will lie in the meta-adaptation support, where best-suited algorithms will be adaptively selected. For this purpose, the Strategy Design Pattern will be extended to the Adaptive Strategy Design Pattern [7]. The Adaptive Strategy Design Pattern defines a self-adaptive strategy. A single strategy referencing the best available concrete strategy is exposed to the client,

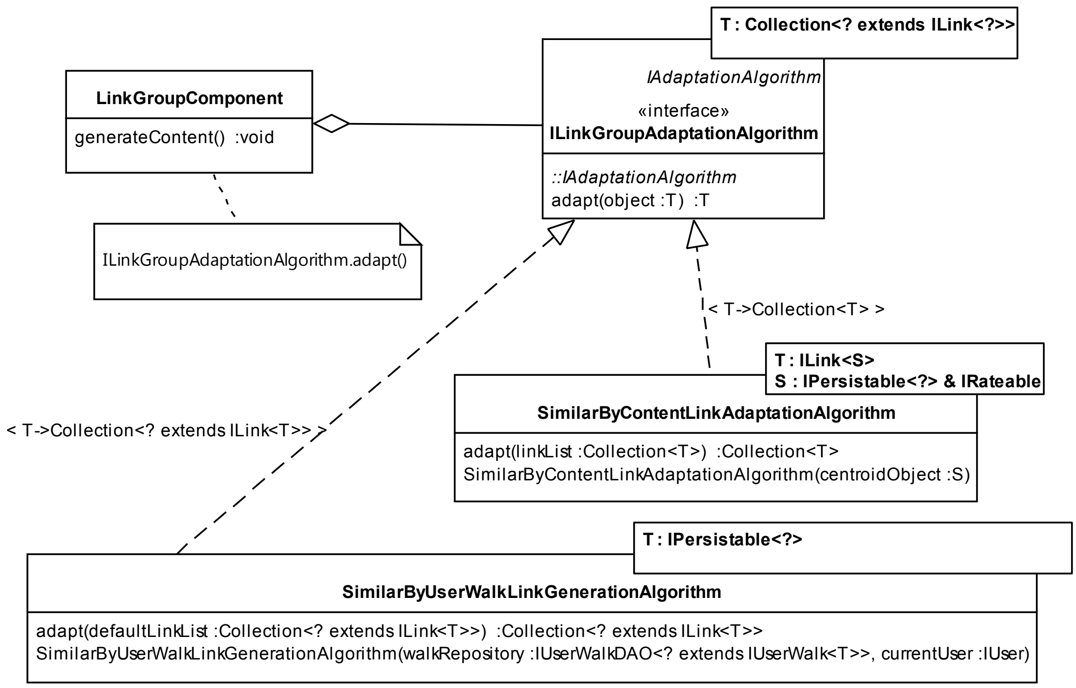


Figure 4.4: Link-group adaptation algorithms.

and the client is required only to provide an access to the environment information that can be used to choose the best strategy.

The data filtering in the user model is based on the conditions (Fig. 4.5). Condition classes follow the Composite Design Pattern [72]. The task of the Composite Design Pattern is to compose objects into tree-like structures to represent part-whole hierarchies. The Composite allows clients to treat individual objects and the compositions of objects uniformly. The conditions have multiple applications in the framework. The same hierarchy is used for evaluating the rules. The purpose of the condition is determined by a particular implementation of the abstract *AdaptationCondition* class.

The rules are defined on the intersections of the user model dimensions. The dimensions are used to limit the information space and to contribute to better evaluation performance. There are two ways of creating the rules in the data storage. The rules can be defined directly by the content designer, or they can be a product of an adaptation algorithm. The combination of these techniques can lead to interesting adaptive behavior. This will be the objective of our future research.

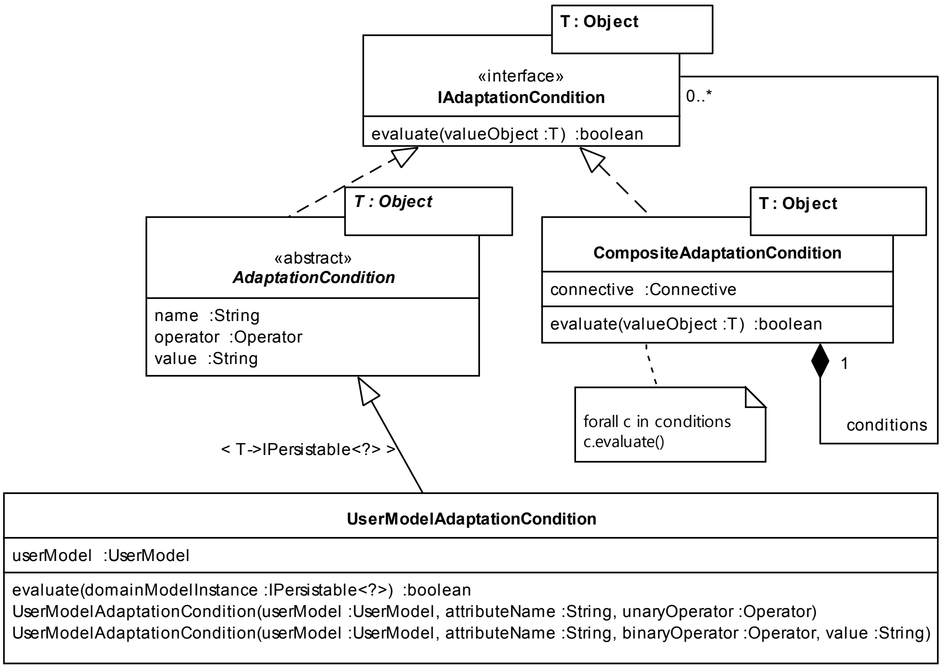


Figure 4.5: Condition class hierarchy.

In the following sub-section, we will show an example of an adaptation algorithm that can be integrated into the system.

Example 4.1 (Walking the document space)**.** To clarify a link adaptation, we will show how the adaptive link generation based on the similarity of navigating paths within the document space is supported by the framework.

Let us have a finite set D of the documents d. A document walk is an ordered set W, where $W \subseteq P$. The document walk is always associated with the user and represents a navigation sequence of the user throughout the document set in a single session. Finding a similar sequence allows us to predict the next most suitable document for a current user based on other users' behavior.

The desired algorithm is classified as a link-group adaptation algorithm based on ASF (Fig. 4.3). In our case, it is a link generation algorithm.

Fig. 4.6 presents a sequence diagram describing typical steps of the adaptation algorithm sequencing. In our specific case, the user walk algorithm, first, requests the default values from the user model (in case of a link generation, this step is not re-

Algorithm: Get possible subsequent documents by comparing the history of the user walk transitions

$document \leftarrow currentWalk$.get($currentWalk$.size$-1$)
for all userWalk in walks **do**
  $newIndex \leftarrow currentWalk$.getPosition($document$)
  $oldIndex \leftarrow userWalk$.getPosition($document$) {compare a similarity of documents preceding the current one in the current and a stored walk}
  while $newIndex \geq 0 \wedge oldIndex \geq 0$ **do**
    if $currentWalkDoc \neq userWalkDoc$ **then**
      break
    else
      increment quality and decrement indexes
    end if
  end while
  if $quality > 0$ **then**
    put into document-quality map
  end if
end for
sort document quality map by document quality
return set of documents sorted by quality

quired), and, second, it loads other user walks from the repository. Based on this data, a set of recommended subsequent documents is returned to the content generator.

4.1.3 User interface

The most user-oriented layer of the framework comprises the user interface components that are customized for the web-page adaptation. The components are based on the JSF and the Primefaces component suite. The components utilize JavaScript and AJAX to provide rich user experience. An added value to the commonly used web components is the tight coupling with adaptive behavior, user model and the adaptation engine. For examples and screenshots, see Section 5.5.

An adaptive text output can be taken as an example of a simple component. In a common web component framework, we can find a text output component generating a text to the web page. The text content selection or customization must be done by the developer. In our framework, we want to provide intelligent web components tightly bound with the adaptation engine. The adaptive text component is able to provide various content adaptation techniques. The functionality of the component should be based on the configuration, selected algorithm and on the provided data storage for the data binding. Any of these parameters can be changed later, without

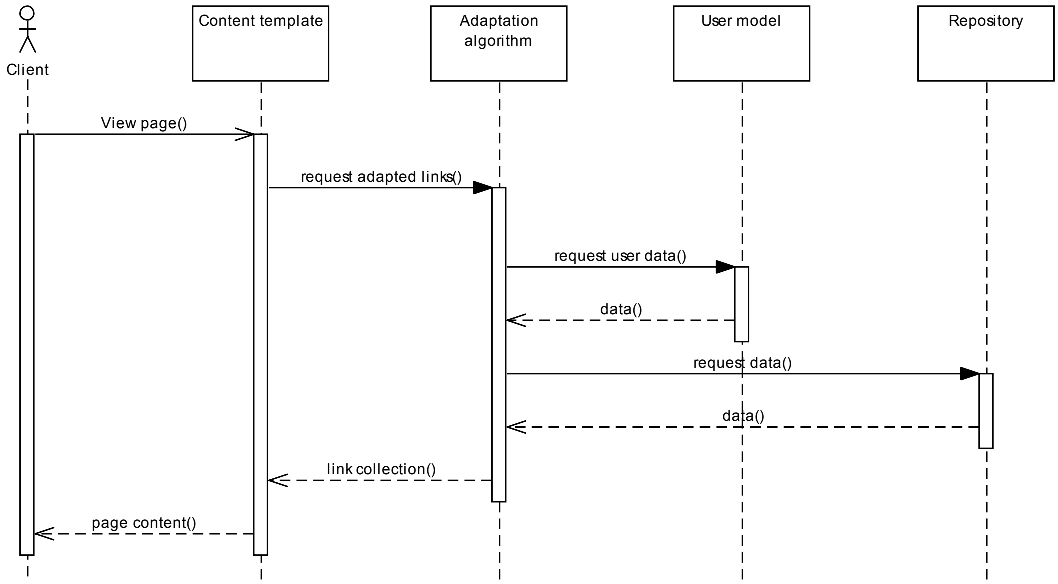


Figure 4.6: Adaptation algorithm usage.

any significant modifications to the web page logic and code.

4.2 AHS Development Process

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. 4.7 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 [109]. 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,

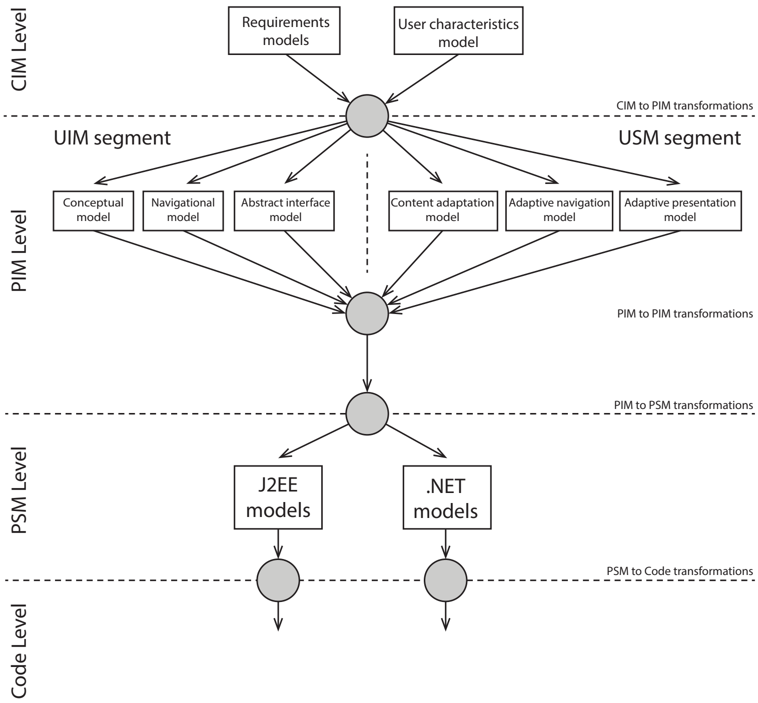


Figure 4.7: MDA structure for user-adaptive hypermedia systems engineering.

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 [102] – content adaptation model, adaptive navigation model, and adaptive presentation model. 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 previous sections, we have proposed a software framework intended to support the development of user-adaptive hypermedia systems. The Adaptive System Framework (ASF) 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 – in case of the learning application called learning 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

The implemented user-adaptive application needs to be evaluated, and various methods mentioned in section 2.8 can be used. In our experiments, we focus mainly on the user-centered evaluation, which is highly appropriate for educational adaptive applications, where a large amount of students can participate.

4.3 Summary

The Adaptive System Framework can be regarded as a possible solution to an effective development of adaptive hypermedia systems. The proposed framework defines fundamental components and is based on a formal model. It provides various possibilities of its implementations and their further extensions. It leads to a simplified process of the development and, at the same time, it does not limit the developer in customized extensions. The default framework implementation is based on both modern frameworks used for the development of the web applications and state-of-the-art technologies of the Semantic Web.

CHAPTER 5

Experimental Verification

Along with our research, we have been trying to apply continuously our theoretical proposals in experimental development. Several use case studies have been implemented. We have applied the adaptive hypermedia and user modeling research in various application areas such as e-learning, e-commerce, social networking and entertainment. The author of this thesis has explored the problem area and has designed a number of simplified use cases. The author has supervised teams of undergraduate students, who have been given practical tasks resulting in working modules or simple adaptive applications.

5.1 Adaptive Learning Case Study I.

This experiment was designed and implemented individually by the author of this thesis as partial evaluation of the proposed AH theory.

Adaptive systems are used in the area of electronic education very often. In the environment of a university, the domain of e-learning creates a lot of possibilities for testing of the adaptive hypermedia systems. That is the reason, why we have selected this area to perform our first experiments.

There is a need of an electronic system for the education support today. It is important for the students to have electronic learning materials, to have permanent access to their results during the semester and to be able to solve assignments online. The next step is the adaptation of the learning materials to the needs and knowledge of particular students. To enable advanced adaptation, the system needs to be based on machine-understandable data representation. Therefore, we have utilized semantic web technologies for this task.

For many areas, including learning environment, suitable ontologies do not exist, or they are not generally applicable. Development of domain ontologies is a topic for another extensive research. Ontologies are usually developed by domain experts and need to satisfy application requirements. A proposal of an ontology development process, that should be followed, was presented in [19].

We have developed a knowledge model for the needs of education practice of the CTU Faculty of Electrical Engineering. For the needs of other universities the knowledge representation might slightly differ, but within the scope of this experiment, we haven't solved the data integration yet. The e-learning ontology is the first step to the ontology suitable for general purpose adaptive systems. The domain ontology provides a possibility to express the semantic of the data. This way, we can later achieve the data interoperability.

The developed ontology consists of two parts. The first is the ontology of course materials, and it includes lecture slides, learning objects used to compose the personalized web page and tasks for students. The second, the ontology of the progress of students in the course, includes test scores, homework scores, evidence of course attendance, etc. These ontologies correspond to two layers of the proposed multidimensional ontology data structure. Subsequent parts are to be implemented in our future work. For the development of the ontology, we use Protégé ontology editor[1]. Figure 5.1 shows part of the ontology.

On top of the ontology, we are developing a web-based tool which will be used to present data to the user. The core of this system has been already implemented in Java language. As a suitable tool for accessing the semantic data, we have chosen RDFReactor[2]. This tool is capable of automatic Java interface generation directly from the RDF schema. The interface consists of Java classes and so the programmer doesnt have to deal with accessing the ontology data. RDFReactor utilizes RDF2GO[3] – an abstraction layer above the data repository. For the data repository access, we use the Jena semantic framework[4].

The designed architecture of the web application (Figure 5.2) corresponds to cur-

[1]Protégé ontology editor `http://protege.stanford.edu/`
[2]RDFReactor `http://semweb4j.org/site/rdfreactor`
[3]RDF2go `http://semweb4j.org/site/rdf2go/`
[4]Jena – A Semantic Web Framework for Java `http://jena.sourceforge.net/`

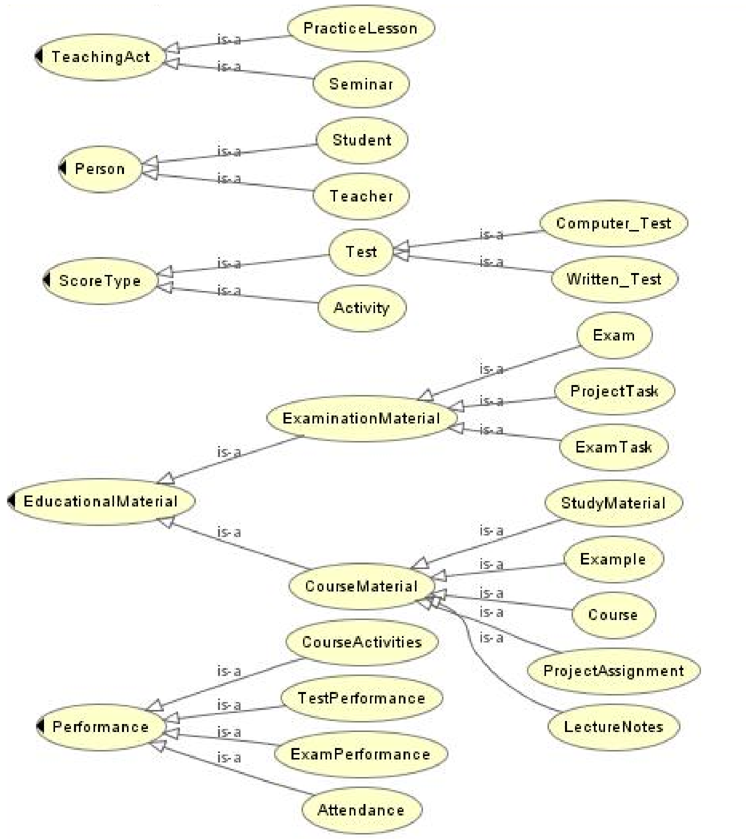


Figure 5.1: Part of our e-Learning ontology in Protégé.

rent trends in the hypermedia systems development. It is based on the Model-View-Controller (MVC) architectural design paradigm, which is recommended for most interactive web applications. MVC makes application functionality more reusable, and simplifies adding and modifying client types, data views, and workflow [147]. Model is responsible for the internal structure of the application, View transforms data and produces the presentation and Controller controls application behavior and user input. This division enables extending and modifying the layers independently.

Graphical representation of technologies used to implement the application is shown in Figure 5.3. The architecture is displayed as a technology stack. The lowest layer represents a database. In this case, we have used PostgreSQL database. Storage layer, consisting of data objects, is based on RDF Reactor and persisting RDF data using Jena into the database. The data objects are part of the model portion of the MVC

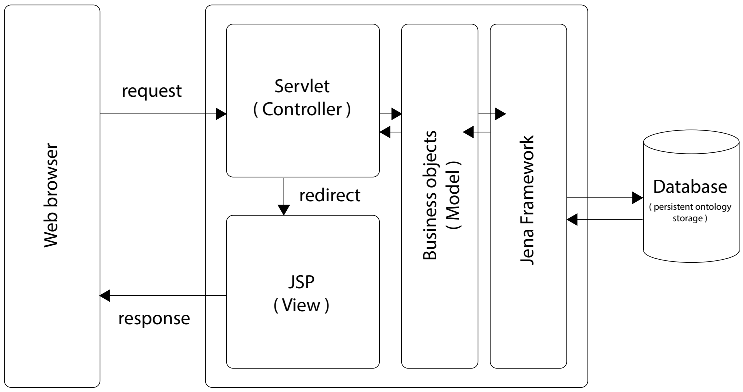


Figure 5.2: Testing framework architecture.

paradigm. JSP technology is the view portion of MVC.

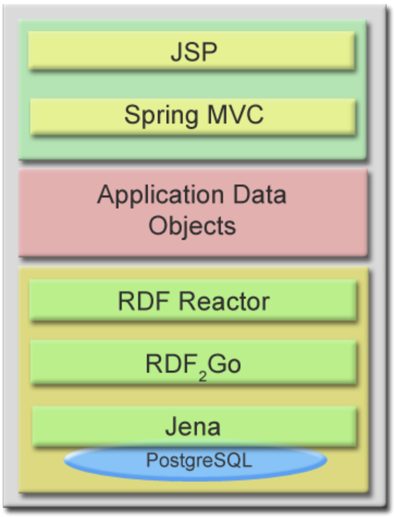


Figure 5.3: Adaptive web portal architecture.

The first adaptive learning prototype has not been applied in real class environment. To evaluate our research, we have performed a study of students expectations from adaptive systems. We have prepared a questionnaire for students using online learning materials of a Java programming course.

The questionnaire studied what properties the users expect an adaptive system to have. The participants were students of the second semester attending the Java programming course. Most of them, especially the beginners, agreed that orientation in current online materials is sometimes difficult. The majority of students would prefer information related to a currently solved task (46%), other students prefer information related to their seminar project (30%) and others to their general knowledge. Most of the students would agree to provide their user model information related to the course to the teacher (66%), or at least statistical results of whole class (25%). This information could be used to improve the program of the lessons.

Achieved results

In the first experiment, we have designed ontology suitable for learning courses at Czech Technical University. Learning courses in this environment will be subject of following experiments. We have designed a suitable architecture using modern web development technologies. We have verified the applicability of semantic web technologies and set up the foundations for subsequent experiments.

5.2 Library and book store adaptation

This experiment was designed by the author of this thesis as partial evaluation of the proposed AH theory. Implementation was realized by Zdeněk Gerlický as part of his thesis [S.1] and was supervised by author of this work.

Based on previous experiences from adaptive semantic learning application, we aimed to apply the similar approach in other application areas. We have designed a complex adaptive hypermedia system with a shared semantic data store.

The main goal of the experimental system was the design of architecture applicable in different domains, supporting adaptability and data storage based on ontology models. Another goal of the implementation was to choose and evaluate technologies used to access the semantic data storage.

Implemented front-ends were from slightly different areas, but both applications work with similar data. One of the applications represents an administration system of a library. Librarian can make entries about borrowed books and reservations. Another system is an electronic book store, where users can search and buy books. Both systems can utilize information about users' genre preferences and favorite books. It is also very advantageous to share this information.

5.2.1 Requirements

The aim of the project is to create a web portal that will observe user's behavior and according to the collected data, it will adapt website content to the needs of specific user.

In the beginning of our work, we have stated several application requirements:

- The system will be a web application.
- It will support adaptive presentation and adaptive navigation based on the user model.
- Two similar data contents will be created within the project and experiments with data integration will be carried out. The data contents will be specified using Semantic Web languages, RDF, RDFS and OWL.
- The system should be platform independent, and therefore, it will be written in Java language.

There were also several architecture requirements based on the previous project:

- The web application will use Spring framework, which enables separation of presentation and application logic by application of the Model-View-Controller design pattern.
- RDF Reactor will be used to transform existing RDF triples to Java objects. An abstraction layer created by RDF2GO provides an interface over a storage layer. Persistence layer will be realized through the Jena framework.

We have identified several other requirements, which had to be followed in the application design. The web application needs to be modular to enable future extensions. The data content should include data semantics and allow reusability and integration. The user interface has to be simple and easy to use. User actions will generate changes in the user model and using the stored information, adaptation will be performed.

5.2.2 Architecture

The system was implemented as a client-server solution with a thin client. Architecture of the system follows the design of the previous project. The architecture is depicted in Fig. 5.3.

The system was built using the application framework Spring, which offers a simple implementation of the MVC design paradigm, resulting in modularity and possibility of future extensions. To fulfill the aforementioned requirements, the data storage of the system is represented by ontologies. In our project, we have chosen the RDF Reactor project as the interface for accessing the RDF storage. RDF Reactor is part of the ongoing project Semweb4j[5] and internally uses RDF2Go and Jena.

As this project was focused on choosing suitable technologies for the development of a modular framework based on the semantic data store and evaluation of these technologies, not all components of the GOMAWE model were implemented. In Fig. 5.4 there are the implemented parts highlighted. This will be in the future extended with implementation of a multidimensional matrix, which will enhance the reasoning process with rules.

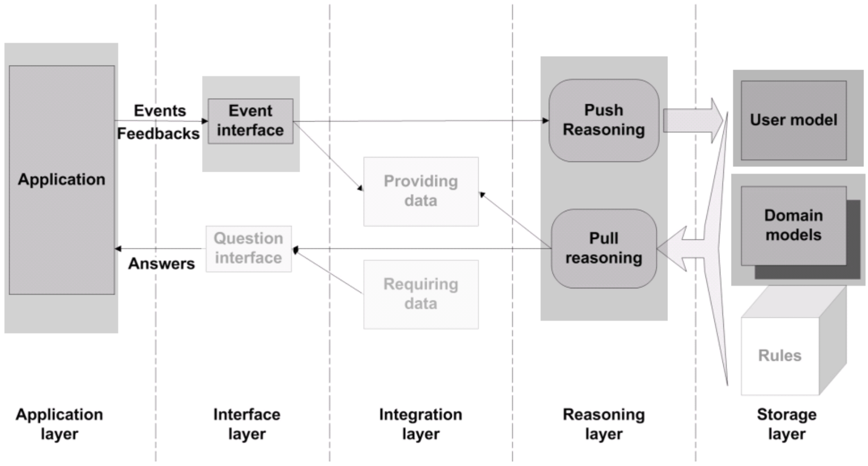


Figure 5.4: Implemented components of GOMAWE.

5.2.3 Semantic data store

In our approach, we use Semantic Web technologies to represent the data by ontology. This innovative approach is not yet supported by the state-of-the-art portal frame-

[5]Semweb4j – `http://semanticweb.org/wiki/Semweb4j`

works. However, there exist some ongoing research projects of adaptive portals based on ontologies [10]. We have created a common ontology for both library and bookstore portals (Fig. 5.5). The common user model resolves the data integration problem and also the problem with user identification.

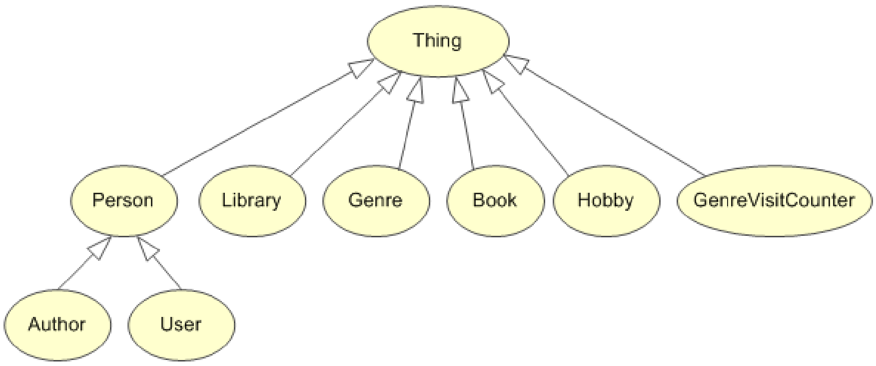


Figure 5.5: Classes in the designed ontology.

The user model contains the following information: books the user has displayed, authors of the books the user has displayed, books selected as favorites, borrowed books, and books the user has bought. These parameters are used in the adaptation process.

5.2.4 Evaluations

First type of evaluation was the acceptance test. This test is typically used in the software development process to verify the functionality. The test was successful.

The second evaluation was the test of the adaptation process. Adaptation is performed based on the user model information. The user model contains initial values given by the preference screen and additional information gained by observing user behavior.

Based on the purpose of the portal, the main subjects of adaptation are books. For each user, the list of books is sorted by the preferences, and the user interface has several possibilities to highlight particular books. Menu items are adaptively sorted as well. The test was performed by defining and evaluating several user behavior scenarios and their influence on user interface adaptation.

We have also evaluated the usability of the application. This test was done by five participants with different computer skills. Before the test, they were asked to fill in some general information about themselves and their computer skills. Then they were asked to perform a simple task in the system and qualify difficulty of the task. After they have finished all tasks, they were asked about general aspects of the application. During the test, some minor issues of the user interface (Fig. 5.6) were detected, but all the participants found the portal easy to use and well graphically designed.

Figure 5.6: Online library user interface.

The important part of the evaluation was the performance test. As we already mentioned before, one of the goals of the implementation was to verify the suitability of selected approaches and software frameworks used. The performance was measured for different amounts of data. We have created four user models of different sizes and tested the adaptation of lists of books with increasing number of books. The measurement has shown that the dependency of processing time on the size of the list of books is linear.

This result was satisfactory for this current application, but for future extensions, we thought it might be improved. The computation time is mainly given by the way the RDF reactor manipulates the ontology data. During the implementation, we also confronted the problem that objects generated by the RDF Reactor do not conform to the Java Bean standards, and the transformation also consumes time. Therefore, we think that some of the queries should be formed in some ontology query language, for example SPARQL. Despite minor problems, the implementation using RDF Reactor was successful, and we have decided to use it in our future experiments.

We have performed a study of student's expectations from adaptive systems. We have prepared a questionnaire for 60 students of a Java programming course. The most important results of the questionnaire are presented in a form of visual representation in Fig 5.7.

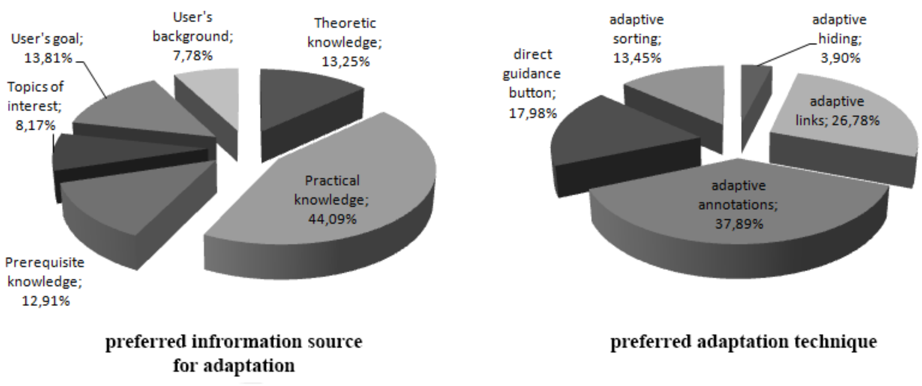


Figure 5.7: User questionnaire results.

From the results, we have concluded, that most of the participants haven't worked with an adaptive system yet. Users expect adaptation of lists of books based on visited or borrowed books by the user. Users also expect adaptation based on favorite books, genres and authors. For setting user preferences, they would prefer a preference screen, where the parameters can be selected. Users expect simplicity of the system and easy to use searching.

Achieved results

As a conclusion we can state that the experimental system was successfully implemented and has been prepared for future extensions leading to the development of the complex semantic based adaptive web system.

5.2.5 ASF implementation

The original application, as it was described in previous sections, was completely refactored and reimplemented by the author of this thesis in the final stage of the thesis project. The aim of the redesigned implementation was to proof the correctness of theoretical GOMAWE concepts and usability of the Adaptive System Framework.

Due to the fact that the original implementation was several years old and was based on outdated technologies, we decided to keep the presentation layer unchanged, and replace other parts completely. Moreover, the RDF Reactor was replaced by Empire, a more recent RDF storage persistence framework. Empire is supported by ASF, and it is currently the main RDF-based persistence implementation of our framework. For more details, see Section 5.6.2.

We have compared the code of both versions of the application. The measures are summarized in Tab. 5.1. From the results, we can conclude, that the code size was significantly reduced with the use of ASF. There are few more classes, which is the result of more precise modularization. It does not mean that these classes were added, but it's a result of quite significant refactoring. Due to the fact, that the presentation layer was not modified, there are additional possibilities for improvement.

Table 5.1: Code size comparison of two Library application versions.

	original implementation	**ASF implementation**
Lines of code	5452	3825
Classes	44	46

Achieved results

We were able to reimplement an application based on relatively outdated technologies to fully match the GOMAWE specification. Except the presentation layer, the aplication is built on top of the ASF framework. This proves the usefulness of the framework and comparison with the previous implementation shows a significant simplification of code.

5.3 Collaborative Filtering Case Study

This experiment was designed by the author of this thesis as partial evaluation of the proposed AH theory. Implementation was realized by Martin Stružský as part of his thesis [S.2] and was supervised by author of this work.

One of the possibilities for adapting information presented to the user is collaborative filtering. One of our experiments focused on collaborative filtering algorithms and their application in adaptive systems. Our approach allows dealing with adaptation techniques as black box components. Therefore, the collaborative filtering library can be used as an adaptation component.

Collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints and data sources. We live in the age of information explosion, and we need tools to process the large amounts of information and offer users the comfort of dealing only with relevant information. The collaborative filtering techniques proved themselves to be very useful for this. Many existing applications use filtering techniques to recommend items such as music [132], books [112], movies, etc. There are many fields where applications benefit from the ability to make qualified recommendations. This could be used even in e-learning to recommend topics appropriate for a users knowledge [117]. Therefore, we think that such type of information adaptation should be part of the adaptive web framework that we are developing.

The collaborative library was designed to be used as part of the reasoning layer of GOMAWE (Fig. 5.8), and was integrated into the ASF. Using artificial intelligence algorithms, we can derive some user characteristics that were not stored in the user model. Similarly, we can add new rules to the multidimensional matrix in the storage layer based on the rules of similar users.

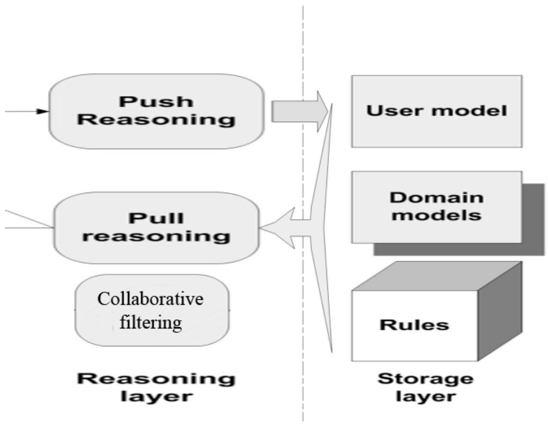


Figure 5.8: Collaborative filtering support in GOMAWE.

5.3.1 Existing applications

Collaborative filtering is based on the assumption that similar users have similar preferences [85]. Recommender systems are typically used in web applications that have many users and want to provide each user with information corresponding to his/her preferences. Amazon.com internet shop portal makes recommendations based on the items users have already bought. Last.fm music portal recommends its users songs based on their recent playlist. Users can also use tags to describe the songs and also to classify them into genre groups. International movie database (IMDb) recommends to the user's movies similar to their favorite ones. These are just the most popular portals, and the recommendation feature is part of many others.

In adaptive web applications, we can also use clustering based on the similarity of users that could help to solve the cold start problem. This means that if we dont have any information about the current user, we could use information about a similar user which we assume to be similar too.

There are many approaches and algorithms to compute the similarity of user preferences [84]. We have selected the simplest and most typically used algorithms, and we will discuss them in the following section. We have implemented these algorithms as a java program and performed experiments with real data.

5.3.2 Selected algorithms

The commonly used algorithm for collaborative filtering tasks is the k-Nearest Neighbor (k-NN) algorithm [146]. The k-NN algorithm is a method for classifying objects based on the closest training examples in the feature space. It belongs to a class of so called lazy learning algorithms. The following formula can be used to calculate the distance d of two users u_a and u_b:

$$d(u_a, u_b) = \sqrt{\sum_{i=1}^{n} (P_a(o_i) - P_b(o_i))^2} \tag{5.1}$$

where n is the number of compared objects o_i and $P(o_i)$ is the rating of the object. After determining the distance of users, we can select K users with the lowest distance and calculate the unknown rating of item i for the user u_0 as the arithmetic mean of rating of the K nearest users:

$$P_0(o) = \frac{\sum_{i=1}^{K} P_i(o)}{K} \tag{5.2}$$

We can also use k-NN algorithm to calculate the similarity of two users as a value ranging from 0 to 1 as:

$$sim(u_a, u_b) = 1 - \frac{d(u_a, u_b)}{\sqrt{n}} \tag{5.3}$$

Another algorithm for determining the similarity of users uses the Pearson correlation coefficient [119]. It ranges from -1 (a perfect negative relationship) to +1 (a perfect positive relationship), with 0 stating that there is no relationship whatsoever. The value of the coefficient can be computed as a quotient of covariance of variables and their standard deviations:

$$r = \frac{cov(X,Y)}{s_X s_Y} = \frac{E((X_i - E(X))(Y_i - E(Y)))}{s_X s_Y} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i - \overline{X}}{s_X}\right)\left(\frac{Y_i - \overline{Y}}{s_Y}\right) \tag{5.4}$$

where $\overline{X}$ and $\overline{Y}$ are sample means and s_x and s_y are sample standard deviations.

Similarity can be also calculated using Spearman correlation coefficient [121]. In principle, ρ is simply a special case of the Pearson product-moment coefficient in which two sets of data X_i and Y_i are converted to rankings x_i and y_i before calculating the coefficient. If there are no tied ranks:

$$\neg\exists i,j : i \neq j(X_i = X_j \vee Y_i = Y_j) \tag{5.5}$$

then ρ is given by:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{5.6}$$

where d_i is the difference between the ranks of corresponding values X_i and Y_i and n is the number of values in each data set.

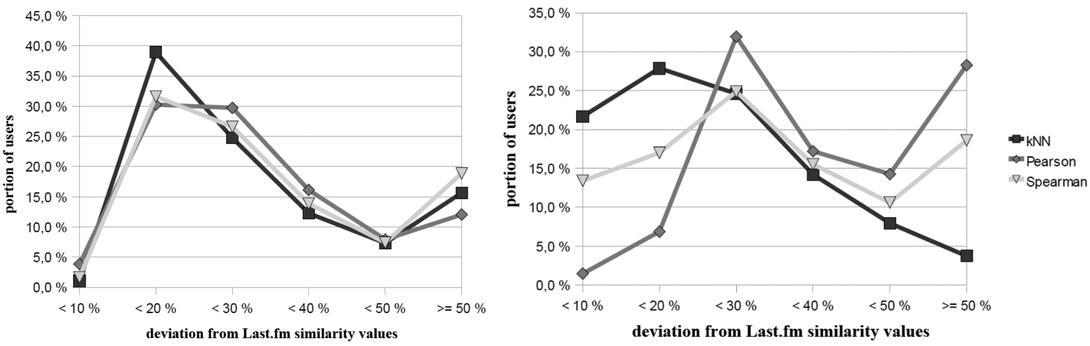
The last algorithm that we used for our experiments uses the Kendall coefficient [1]. Kendall τ coefficient is defined as:

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{5.7}$$

where n_c is the number of concordant pairs and n_d is the number of discordant pairs.

5.3.3 Experiments

We performed experiments in the field of music recommendation. We used our library with implemented algorithms that we explained in the previous section, and we used data from the music portal Last.fm. We have analyzed data of five users and compared



(a) (b)

Figure 5.9: Portion of users depending on the deviation from Last.fm similarity values.

the results of our algorithm implementations with the results of the Last.fm service. To be able to compare the data, normalization was needed. All values were multiplied by a coefficient. This coefficient was chosen so that the highest value of similarity computed in our library, and the highest value reported by the Last.fm service were equal. All negative values were set to zero. This is because used input data contained 50 interprets with the best rating from the selected user. Therefore, we have no possibility to determine, which items had the worst rating. However, in some other cases, even negative values could be used to reason about user preferences.

In Fig. 5.9. there are the results for two selected users. The values were computed using three of the algorithms, and we observed the deviation from Last.fm similarity values. Compared to processing all data, we achieved better results by eliminating users with similarity values near zero.

We achieved the best results with the k-NN algorithm. The number of processed users has significant influence on the results. Optimalization could be also achieved by changing the k value number of nearest neighbors.

The algorithms were implemented as a java library. For the experiments, we used a graphical user interface. However, the library could be used separately, e.g. in the adaptive web portal backend. This will be the next challenge, and we will perform experiments corresponding to the scheme of the GOMAWE that we mentioned in the text earlier.

Achieved results

This experiment has been focused on the reasoning layer of GOMAWE model. This layer is the place, where user modeling and information retrieval is performed. Our design assumes adaptation algorithms as black box components, that could be extended or replaced easily. In the experiment, we have implemented a library of collaborative filtering algorithms, that served as evaluation of our approach.

5.4 Integration using web services

Adaptive Hypermedia Systems observe users' behavior and provide personalized hypermedia. Users interact with many systems on the Web, and each user-adaptive system builds its own model of user's preferences and characteristics. There is a need to share the personal information, and the current research is exploring ways to share user models efficiently. In this section, we present our solution for personal data exchange among multiple hypermedia applications.

5.4.1 SOAP Web service prototype

This experiment was designed by the author of this thesis as partial evaluation of the proposed AH theory. Implementation was realized by Lukáš Koranda as part of his thesis [S.6] and was supervised by author of this work.

For evaluation of integration features of adaptive hypermedia systems, we have designed a project for simple semantic data synchronization. We have created data based on the Friend of a Fried (FOAF)[6] ontology. The aim of the FOAF project is creating a Web of machine-readable pages describing people, the links between them and the things they create and do. It is an open, decentralized technology for connecting social Web sites.

Our experiment demonstrates semantic data synchronization using a SOAP web service. The web service provides an interface to a personal contact storage. The storage structure is defined by the FOAF ontology. The main storage was implemented as a central server, that offers synchronization with multiple clients. The important components of the experiment are depicted in Fig. 5.10.

Client application has its own storage. That could be a local database or simply a file. For synchronization with central storage, the client calls web service methods provided

[6]The FOAF project `http://www.foaf-project.org`

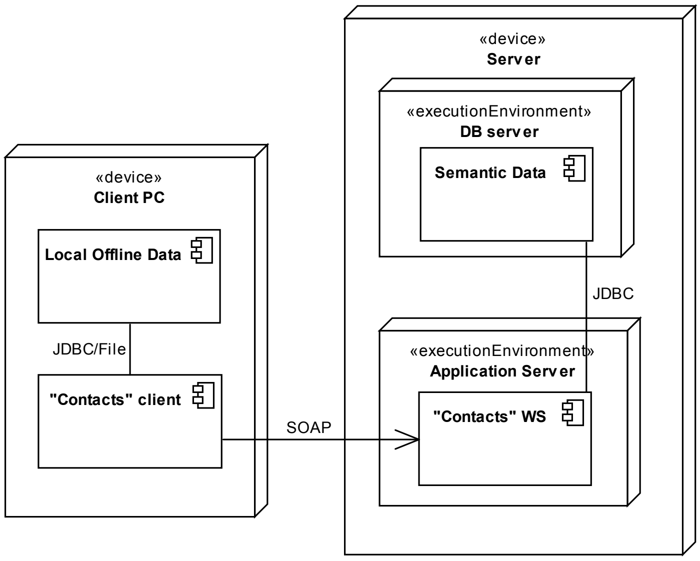


Figure 5.10: Deployment of the "personal contacts" project.

by the central server.

The required operations were following:

- Adding new contacts into the storage
- Updating existing contacts in the storage
- Removing out-dated data
- Retrieving data from the storage based on its URI
- Suitable interface for searching in data

Important aspect of the communication is the synchronization feature. Adding new contact adds a new record only if it does not exist. Otherwise, new properties are updated in the existing record. Update operation also removes record properties that were removed on the client. Record removal affects only the assignment to user's known contacts if the record is referred by others.

Our aim was to design a generic web service that could be used for the exchange of any data. The possible solution was to represent the data in RDF/XML format. In that way, the web service could be used in any following experiments and can send data stored in the semantical data storage.

Achieved results

Information exchange is the primary purpose of the integration layer of GOMAWE model. In the described experiment, we have implemented and evaluated the communication interface in a form of a SOAP web service. In this project, the server and clients, all worked with the same ontology (FOAF) and we did not solve the ontology mapping issues. That may be addressed in future work. Additionally, the communication interface could be provided as REST service in future versions.

5.4.2 REST Web service prototype

Representational State Transfer (REST) [68] architectural style was proposed by Roy Fielding and is based on a set of principles for designing network-based software architectures. The set of principles was defined by four interface constraints: identification of resources, manipulation of resources through representations, self-descriptive messages, and hypermedia as the engine of application state.

Web services based on the principles of REST are called RESTful and can be considered as an alternative to the SOAP-based web services. In the past, RESTful services were used only for simple ad-hoc services, and the area of enterprise systems was scoped to the WS-* standards, e.g., SOAP, WSDL, WS-Addressing, WS-Security. This is no longer the case and RESTful services have been successfully applied in many enterprise applications. The challenge is to use them correctly and to be able to align them to solve the real problems [2]. The comprehensive comparison of both technologies was presented in [137, 136]. Conclusions of the comparison give an advantage to the RESTful services in Web integration scenarios and prefer WS-* Web services for enterprise application integration, where advanced security and Quality of Service (QoS) is required. The practical comparison in [86] concludes that REST is lightweight, scalable, very easy to understand, learn, and implement.

There are two important terms when speaking about REST methods – safety and idempotence [70]. Safety means that calling the method does not cause side effects and does not change the state of the server. For example, the `GET` method of an API must adhere to the safety definition, otherwise it can cause problems for other services and result in unintended changes on the server. Idempotence refers to a method that will produce the same results if executed once or multiple times. The `PUT` and `DELETE`

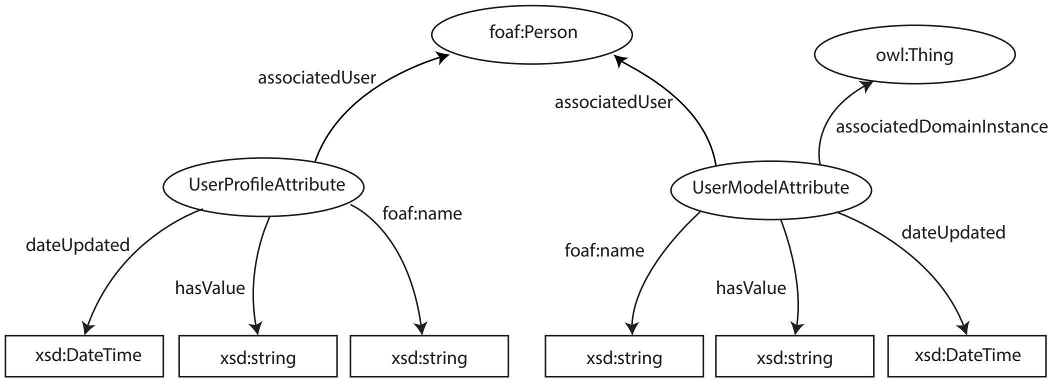


Figure 5.11: Selected parts of the GOMAWE Architecture Ontology.

methods are defined to be idempotent. Therefore, it should be ensured that making multiple requests produce the same result on the server. Safe methods are idempotent at the same time, because they do not cause any changes on the server.

A RESTful user-specific data interface

Based on the comparison in the previous subsection, we decided to use RESTful web services to design the user-adaptive application's personal data interface. First, the interface does not require advanced security, and second, the most important requirement is flexibility supplemented with easy and intuitive development.

The exchange of information is based on standard metadata vocabularies and ontologies. Utilizing the proposed unified user model data structure, ontologies can be aligned with relative ease and translation between two domains can be achieved. At the same time, the design does not force participating systems to agree fully on a fixed domain model ontology, and advanced mediation techniques can be used when needed.

RDF vocabularies

Our user-data-integration interface uses three types of vocabularies – first, the standard vocabularies, second, user-model-specific vocabularies and finally, the domain-specific vocabularies. RDF resources and their attributes reuse existing, widely adopted vocabularies such as the Dublin Core [7], Friend of a Friend [8], and Semantically-Interlinked

[7]Dublin Core Metadata Initiative – `http://dublincore.org/`
[8]FOAF vocabulary – `http://xmlns.com/foaf/0.1/`

Online Communities [9] vocabularies. To ensure good interoperability, ASF maps as many attributes as possible to these standard vocabularies. User-model-specific vocabularies used in our design include IntelLEO User Model Ontology [10] and GOMAWE Architecture Ontology, see Fig. 5.11. Domain-specific vocabularies differ application from application and refer to domain-specific concepts to which the user's characteristics are related.

The Table 5.2 lists the vocabulary namespaces used in the ASF-based application REST API.

Table 5.2: Vocabulary namespaces used in the REST API.

Prefix	**Namespace URI**	**Description**
dcterms:	http://purl.org/dc/terms/	Dublin Core vocabulary
foaf:	http://xmlns.com/foaf/0.1/	Friend of a Friend (FOAF) vocabulary
sioc:	http://rdfs.org/sioc/ns#	Semantically-Interlinked Online Communities (SIOC)
owl:	http://www.w3.org/2002/07/owl#	Web Ontology Language (OWL) vocabulary
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF vocabulary
rdfs:	http://www.w3.org/2000/01/rdf-schema#	RDF Schema vocabulary
xsd:	http://www.w3.org/2001/XMLSchema#	XML Schema (XSD) vocabulary
um:	http://intelleo.eu/ontologies/user-model/ns	IntelLEO User Model Ontology
gomawe:	http://fel.cvut.cz/gomawe/0.1/	GOMAWE Architecture Ontology

ASF-based application API resources

The fundamental resources of the RESTful API are listed in Table 5.3.

Table 5.3: Primary resources of the REST API.

Type	**Example**
gomawe:UserProfileAttribute	/api/users/{user_id}/profile
gomawe:UserModelAttribute	/api/users/{user_id}/model
foaf:Person	1) /api/users/{user_id} 2) /api/users/lookup?email={emailAddress}

In Table 5.3, `gomawe:UserProfileAttribute` refers to the list of User Profile attributes assigned to the user with the matching ID. `gomawe:UserModelAttribute` refers to the list of User Model data of the user with the matching ID. `foaf:Person` refers to the owner of the user account corresponding to the specified id (1) or email address (2).

[9] SIOC Project – `http://sioc-project.org/`

[10] IntelLEO User Model Ontology – `http://intelleo.eu/ontologies/user-model/spec/`

REST operations

This subsection summarizes the REST operations supported by the API. The API supports three HTTP operations: `GET, PUT, POST`. The `DELETE` operation is not supported, as the user data can only be extended or updated, and no attributes can be removed.

Only the RDF/XML format is supported at this time, therfore the HTTP Accept header value should be set to:

```
GET /api/users HTTP/1.1
Accept: application/rdf+xml
```

The following parameters can be used to refine the requests:

- Modified since – Parameter `modifiedSince` can be used to limit the listed attributes to only those that were modified after the entered time value. The parameter value is of type `xsd:date` or `xsd:dateTime`
- Resource paging – Parameter `limit` can be used to limit the number of listed attributes to avoid large data transfers. Parameter `offset` can be used to request additional pages starting with the specified item. For both parameters integer type value is allowed.

The following are the typical uses of HTTP methods with explanatory examples:

- HTTP GET – The `HTTP GET` method is supported by all API resources. It is used to retrieve the user's account and his or her user modeling data. First, the application needs to negotiate the correct resource IDs and after matching user's account, the user model attributes can be requested, see Fig. 5.12. The following request can be used to retrieve user profile attributes of a user:

  ```
  GET http://example.org/api/users/
     {user_id}/profile HTTP/1.1
  Accept: application/rdf+xml
  ```

- HTTP PUT – The `HTTP PUT` method is supported only by UserProfile and UserModel resources. It is used to update the models by an external application.

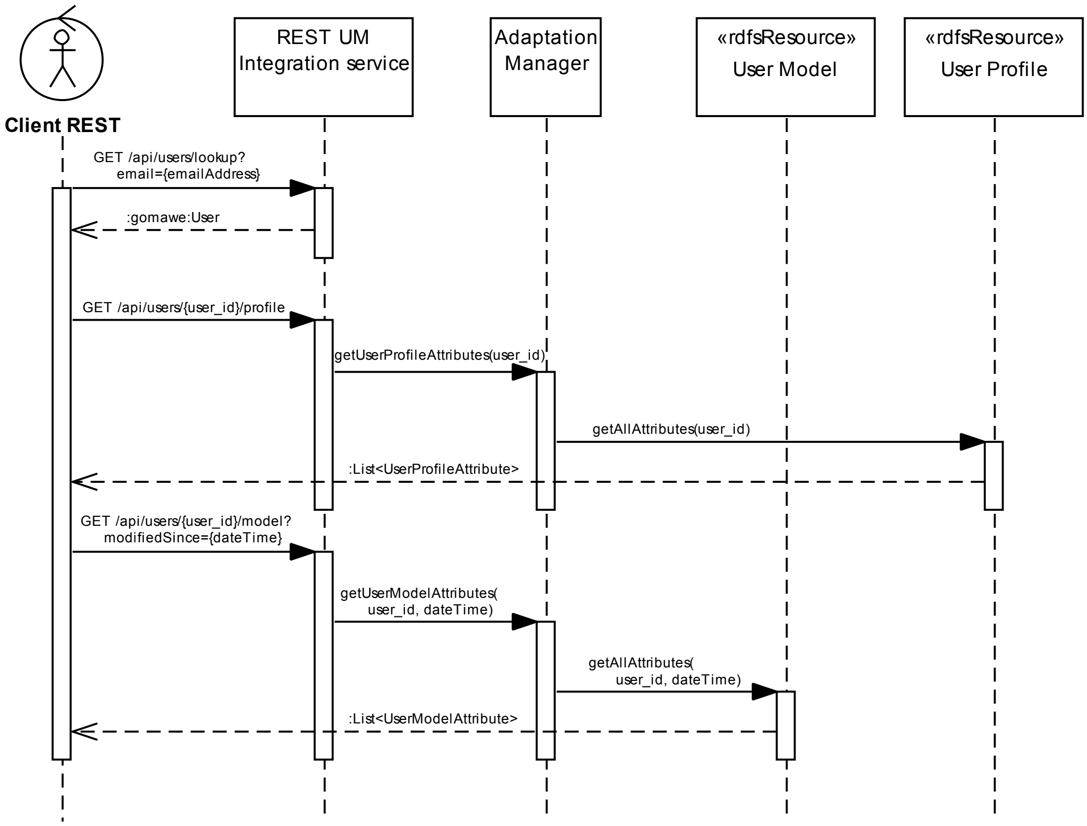


Figure 5.12: Example of request sequence to retrieve users's personal data.

The `PUT` method operation needs to be idempotent, therefore, using this method means creating new attributes or replacing values of existing attributes.

```
PUT http://example.org/api/users/
   {user_id}/profile HTTP/1.1
```

- HTTP POST – The `HTTP POST` method is supported only by UserProfile and UserModel resources. The operation of this method is reserved for the cases of incremental updates of the models. It will be more extensively utilized in future extensions of the REST API. The more precise value updates can be based on arithmetic mean or the time of last update. This operation is not idempotent and should be called only once to avoid inappropriate user model changes.

Authentication

To avoid misuse of both the personal and the domain data provided by the RESTful application interface, authentication of requests is required. We use HTTP basic authentication. Using HTTPS protocol is recommended. Otherwise, the username and password would be sent without encryption.

Application use-case

The proposed approach of AHS integration will be demonstrated in an adaptive learning scenario. There are currently three separate systems used by students in a programming course, each with a different purpose. Although each of the systems stores different information, they can benefit from each other, exchange user's data and extend understanding of the user's knowledge and preferences.

The integration module implementation is a work-in-progress project, and the results will be evaluated in the future. Currently, we do not focus on ontology mapping issues. Our aim is to verify the web service interface and the method of exposing user-specific personal data.

There are three participating information systems in our scenario. The first system is an adaptive learning system containing learning materials and simple test questions at the end of each lesson. The second system is intended for selection of a programming project topic and for submission of the completed works. The third system performs an automatic evaluation of several programming tasks assigned to students in the course of a year. The adaptation features of the learning system can benefit to a considerable extend by additional information about students's progress on solving the assigned tasks and their achieved results. The adaptive guidance within the learning course can be also really well tailored based on the student's project topic, and the subtopics related to his or her project.

In the current setup, see Fig. 5.13, each of the systems is equipped with the RESTful data interface, and all systems are interconnected through a central mediation service. Each of the systems conforms to a similar domain, and their domain models overlap with some mutually related concepts.

The User Model Attributes exchange is subject to a certain level of mutual understanding of the domain semantics by the communicating counterparts. In our experimental scenario, for the ease of personal-data interface demonstration, all three systems use the domain ontology of the adaptive learning system. The data structure defined by

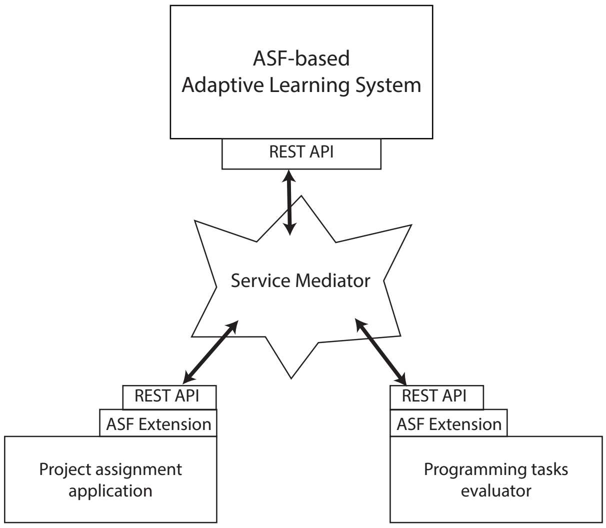


Figure 5.13: Applications participating in the use-case integration scenario.

the ontology is mapped locally. Furthermore, the RESTful API of the learning systems is extended by the domain resources and systems can negotiate the correct domain instance references. All domain instances are represented as RDFS-resources identified by a unique Uniform Resource Identifier (URI). The URIs are used as a link between the overlay User Model and the domain layer of the application.

The User Profile Attributes are not domain-dependent. The meaning of the included preferences needs to be well represented using common and widely used ontologies. In later implementations, public services like Facebook, Twitter or LinkedIn can be used as user's personal information providers, and therefore, it is important to be able to match the preferences adequately.

Achieved results

In this experiment, our novel solution to deal with the syntactic and semantic heterogeneity of personal information in adaptive hypermedia systems has been proposed. To make the integration of users' preferences and characteristics possible, we utilized the GOMAWE specification, and a RESTful web service application interface was introduced.

Compared to other solutions presented in Section 3.1.6, our proposal combines both shared format and conversion approaches resulting in a hybrid solution and utilizing advantages of both approaches.

The strength of our approach is the generic architecture, and the fact that realization can be supported by the Adaptive System Framework. Even non-adaptive systems can be extended by the ASF-based integration module and operate as providers of users' personal data. An integration use-case of that kind was presented using the case of multiple learning-support applications. As a result, the adaptive learning application can benefit from the integration and promptly react on achievements of students recorded by related learning-support systems.

5.5 Adaptive UI components

This experiment was designed by the author of this thesis as partial evaluation of the proposed AH theory. Implementation was realized by Ondřej Harcuba as part of his thesis [S.8] and was supervised by author of this work.

The aim of this experiment was to create a software library of web components, that support adaptivity and can be easily used by a web-site developer.

Based on our experience from previous projects, we have chosen to build the adaptive components on top of Primefaces[11] component library. Our previous attempts were based solely on JSF and jQuery and did not bring sufficient and reusable results. The problem was the complexity and browser-compatibility issues of the non-adaptive portions of the components. The Primefaces component library helped us to focus especially on the adaptation solutions.

Starting point for the design of adaptive components was the analysis of the adaptation techniques' taxonomy (Fig. 2.1). The techniques of our interest were especially from the categories of adaptive presentation and adaptive navigation techniques. We did not focus on the content adaptation techniques, because they are tightly bound to user model and adaptation component. This requires integration with other layers of the ASF project, which was planned for the next research phase.

We have implemented the following components:

- **Rated link**
  Rated link is a type of adaptive annotation component, which is categorized as adaptive presentation and partially as adaptive navigation. Adaptive annotation is very powerful technique and is more effective than adaptive hiding, because

[11] http://primefaces.org/

annotation can distinguish more than two states [23]. Annotations are provided in the form of visual cues. Usually link annotation with different icons, color or font size is used.

Our demonstration of the component (Fig. 5.14) differentiates four states - suitable, unsuitable, visited and blocked. Tooltip with additional information appeared next to the link when the user moused over it. This is another annotation possibility.

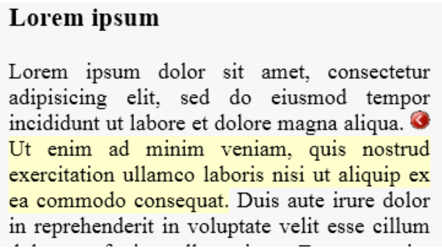
Figure 5.14: Rated link component.

- **Rated link tree**
  Specialized tree component is used to group rated links according to their hierarchy. This component is particularly useful for menu or web page guidepost.

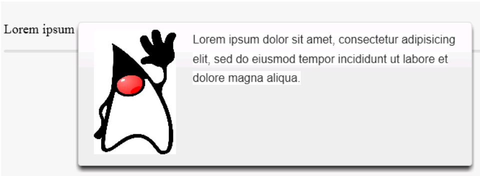
- **Accordion text area**
  Adaptive variant of the text area component transforms specifically annotated parts of text to a roller (Fig. 5.15a). Text can be hidden or shown by clicking on an icon. This is adaptive presentation technique. The default state can be determined by the adaptation component.

- **Overlay text area**
  Another type of presentation adaptation is overlay text area (Fig. 5.15b). This component displays small panel with additional information, or with related links, next to a keyword when hovering the mouse over the keyword.

- **Time monitor**
  The time monitor component does not have any visual representation on the web page, but it is primarily specialized on adaptation process support. This component reflects the well-known issue of AHA! and some other existing adaptive applications – if the user clicks through all pages, even without reading them, the content is assumed to be known. This component monitors the amount of time spent on the page. To be more precise, user's actions are monitored through



(a)



(b)

Figure 5.15: Accordion text (left) and overlay text (right) controls.

mouse movement and keystrokes, and idle time is ignored. The measured time is then used for an enhanced user model update.

- **Guidance page map**
  To enhance the navigation possibilities in a subset of hypermedia pages, we have designed the guidance page map component (Fig. 5.16). This component extracts the links from a page, determines their relations and displays the pages as an oriented graph. Nodes are represented by the "rated link" component, which allows additional annotations. Orientation of edges is determined by page prerequisites. Based on the prerequisites, the component can compute the best path to a selected page, or can determine the best next page for the user.

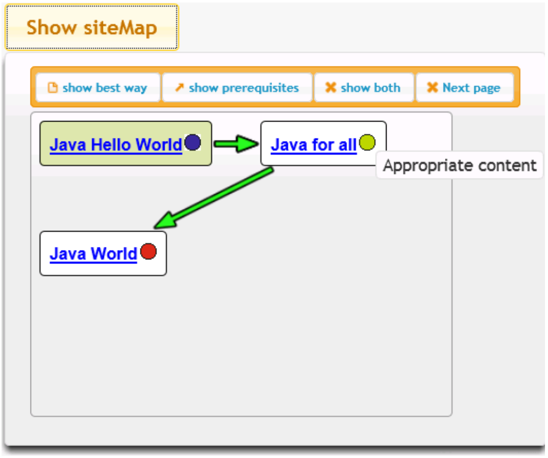


Figure 5.16: Guidance page map component.

Next two components are intended for the authoring counterpart of the adaptive system.

- **Accordion editor**
  Extended version of HTML editor was created as a counterpart for the "accordion text area" component. The toolbox of the editor was extended with commands for adding the code delimiting collapsable text area. For additional components that would depend of content annotations, it would be appropriate to extend this editor too. It is necessary, that the content author has a support for adding adaptation features.

- **Concept creator**
  For editing concepts and their relations, concept creator component was designed. It is a javascript-based editor, where the adaptive system designer can create a tree hierarchy of concepts. An emphasis was put on domain independency and reusability of the component.

Achieved results

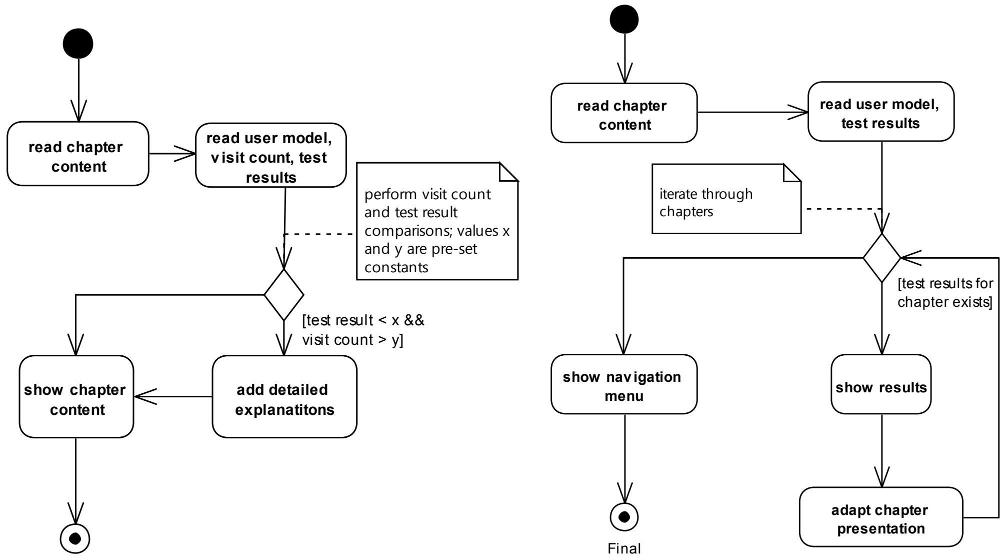
We can conclude, that this project was successful in finding and evaluating suitable base support for adaptive components' development. Examples of most typical adaptive components were created. The most important achievement was, that the development method can be reused in subsequent projects and growing set of reusable adaptive components will be provided by ASF.

5.6 Adaptive Learning Case Study II.

Before we describe the last implemented adaptive system prototype, we should mention a preceding learning course implementation. It was a music theory course implemented by Michal Snížek [S.7]. It was a simple application capable of adapting chapter menu annotations and hiding of content fragments based on user's test results. The main adaptation workflows of the application are depicted in Fig. 5.17. This experiment served as preliminary evaluation of the fundamental principles of adaptation in a learning system and the results were utilized in the subsequent, more advanced project.

5.6.1 Adaptive Java course

This experiment was designed by the author of this thesis as partial evaluation of the proposed AH theory. Implementation was realized by Miroslav Hořejší as part of his thesis [S.9] and was supervised by author of this work.



(a) (b)

Figure 5.17: Adaptation workflows in the music theory tutorial.

To validate our theoretical proposals, we have implemented adaptive learning system that is based on the selected fundaments of the GOMAWE model. This demonstrates an application of the model in the learning environment. However, the general model is applicable to any hypermedia web-based system. Therefore, the modeling of personalized learning can be added as an extension.

An important aspect was to separate the adaptive application and the authoring tool (Fig 5.18).

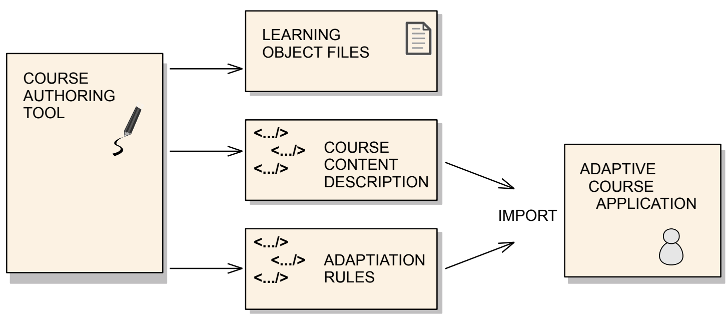


Figure 5.18: Interface between authoring tool and adaptive application

The data exchange between these modules is realized in three separate data structures. The atomic learning objects (texts, pictures, code snippets, etc.) are referred as web resources and identified by URL. In our simple prototype, the data were stored as files in a shared directory hosted by a web server. The course content hierarchy and the layout of elements are in the prototype described by an xml document. An example of this document is shown in (Listing 5.1).

Listing 5.1: Adaptive course content hierarchy and layout

```
<adaptiveCourse>
  <lessons>
    <lesson>
      <chapters>
            <chapter>
          <name>First Chapter</name>
          <number>1</number>
          <pages>
            <page>
              <number>1</number>
              <rootNode xsi:type="borderLearningLayout">
                <top xsi:type="textLearningObject">
                  <xmlId>ref1</xmlId>
                  <url>http://host:port/course/01-01-01.txt</url>
                </top>
                <center xsi:type="codeLearningObject">
                  <xmlId>ref2</xmlId>
                  <url>http://host:port/course/01-02-02.java</url>
                </center>
              </rootNode>
            </page>
            <page>
              <number>2</number>
              <rootNode xsi:type="textLearningObject">
                <xmlId>ref3</xmlId>
                <url>http://host:port/course/01-01-02.txt</url>
              </rootNode>
            </page>
          </pages>
        </chapter>
      </chapters>
    </lesson>
  </lessons>
</adaptiveCourse>
```

Additional xml document contains content adaptation rules as they were defined by the course author (Listing 5.2). The data structure correcponds with the Inference Rule definition, see Def. 3.13. The theoretical model assumes data representation using semantic web languages, e.g., RDF and OWL. This is necessary for more complex reasoning planned in the future. However, by using suitable object mapping, the XML data format can be easily replaced by ontology.

Listing 5.2: Content adaptation rules

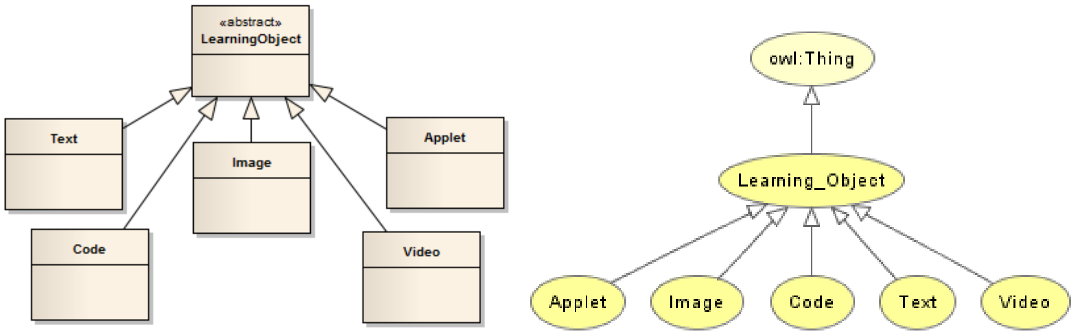
```
<contentAdaptation>
  <rules>
    <rule>
      <condition>
        <name>lesson_1_test_result_percentage </name>
        <operator>LESS</operator>
        <value>50</value>
      </condition>
      <refId>ref1</refId>
      <url>http://host:port/course/text2.txt</url>
    </rule>
  </rules>
</contentAdaptation>
```

We have applied the Adaptive System Framework in a learning course development. We have chosen an adaptive learning environment, because adaptation is very often applied in this area. The university environment provides various possibilities to evaluate such application in courses when used by students.

The domain of the experimental learning system is persisted in a relational database. The implementation of the storage layer was based on Java Persistence API (JPA) and Hibernate implementation. However, our model is designed to comply with ontologies, to maintain data in a triple store and allow SPARQL queries. Therefore, we show the learning object hierarchy (Fig. 5.19) both as classes and ontology. This choice of storage has been limiting some benefits of the framework. However, it was sufficient as a prototype to validate the framework architecture. In future extensions of the adaptive application, ontologies will be used, data will be saved in a triple store and integration features will be evaluated.

In the current version of the application, only basic adaptation techniques are implemented. Practical knowledge is implemented in a form of tests. Theoretical knowledge is presumed after accessing and reading the page and prerequisite knowledge is given by the relation of topics.

In Fig. 5.20, you can see sequence diagram describing interactions of components during the adaptation process. When a user enters a page, the related topic is labeled as read and the information is stored in the user model. Similarly, when the user passes an assessment related to a topic, the topic is labeled as known.



(a) (b)

Figure 5.19: Domain - Learning objects.

The adaptive system was tested with students of a programming course. They studied one chapter of C language basics that contained four lessons. Each lesson was concluded with a test. Test assessments were taken into consideration while recommending the most suitable content to learn.

The application was tested by 35 students, and the content was limited to one topic of one week of the semester. From the results of the log analysis, we could get interesting observations regarding the feedback from the students. In this phase, the feedback was limited to a preference screen only. While using an online course, 5 students (14%) tried to change the adaptation setting and 5 of them tried to reset the result statistics. More students were interested in personal settings, particularly, the visual theme of the application. 12 students (34%) changed the visual theme. From these results, we can conclude that a default setting of the adaptation is very important and that the adaptation based on automatic observations of the users should be extensively applied. An explicit feedback can be expected after the users become more friendly with the system and start customizing the system to be more comfortable to use.

After finishing the adaptive course, students were given a questionnaire where they were asked to rate the tutoring system in a number of questions by grades (1 was the highest and 5 the lowest grade). They were also asked to give an optional feedback.

We can deduce some interesting results from the ratings of students (Fig. 5.21) and their additional comments. Most of the students found the adaptive features of the system useful.

Some of them did not notice the adaptive features at all. However, it is not important

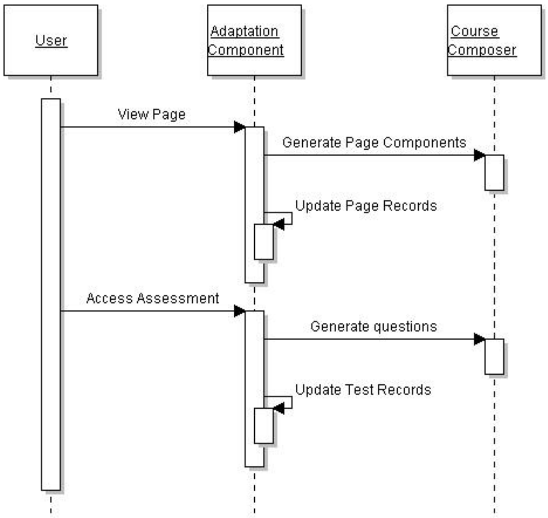
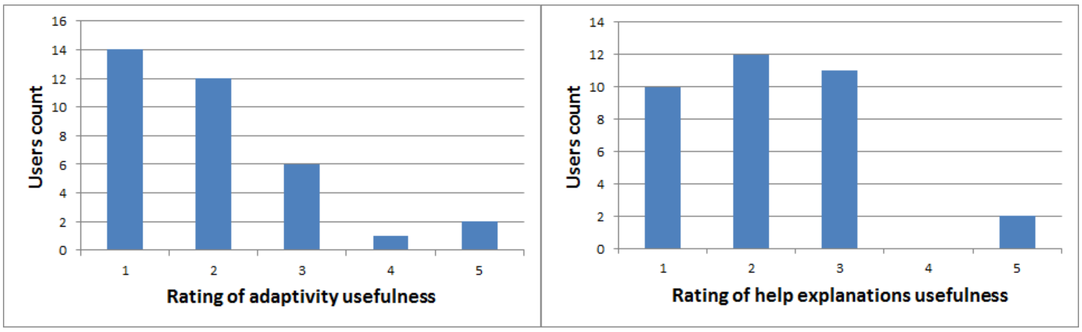


Figure 5.20: Basic adaptive interaction

for users to be aware of an adaptation, but to be comfortable with the system features.



(a) (b)

Figure 5.21: Student questionnaire results. Rating ranges from useful (1) to useless (5).

Several comments mentioned display issues on netbooks with low resolution screens. From this, it is obvious that users context needs to be taken into account and optimize the presentation for the available screen resolution. Some students also did not like the style of presentation where lessons were split into multiple simple pages with arrow navigation. These students would prefer a scrollable page covering the whole topic. This could be an additional option in the users preferences.

We have received also some interesting comments regarding tests used to evaluate knowledge progress of students. Because the tests in current application are statically defined for each lesson, some students pointed out that the questions could be randomly selected. This is where adaptation could also be used. Questions could be selected based on students knowledge and learning progress.

Achieved results

The implemented adaptive learning system prototype was fully functional, and it was evaluated by students while using the online course. However, the requirements were intentionally simplified and did not fully correspond with all features of our theoretical model. On the other hand, our model is generic and applicable for the development of any type of AH application. This was partially proven by this experiment.

5.6.2 Semantic extension

The work on the adaptive learning application has continued. The subsequent experiment was designed by the author of this thesis to implement the GOMAWE model in most aspects as possible, and to provide the evaluation of the proposed AH theory. Implementation was realized by Jan Fiala and was supervised by author of this work. The student will submit his thesis in January 2014.

The most significant change in the architecture was the replacement of database storage layer with a triple store. We used the fact, that the original application was written using JPA and its Hibernate implementation. We did a research on suitable triple store mapping tools. We were looking for a tool providing the same interface for semantic triple stores. As a best candidate satisfying this requirement, Empire[12] project was chosen. Empire is an implementation of the Java Persistence API (JPA) for RDF and the Semantic Web [82].

Empire implements as much of JPA as possible to provide access to a SPARQL endpoint. User of the interface is abstracted from the RDF details, which simplifies the use by new developers. For our purposes, we needed to integrate the fundamental functionality into ASF. Full implementation of JPA for semantic data is impossible. Therefore, Empire is more like an extension of the API. Because there are many implementation details different, we created a special module of ASF for the Empire implementation.

Mappings between classes and storage are controlled through annotations. In the following code, you can see the typical JPA *@Entity* annotation. It is extended

[12] https://github.com/mhgrove/Empire

by Empire-specific annotations providing the name of a mapped RDFS class and namespaces to allow using qnames instead of full URIs.

```
@Entity
@Namespaces({Gomawe.PREFIX, Gomawe.NAMESPACE,
             Foaf.PREFIX, Foaf.NAMESPACE})
@RdfsClass(Gomawe.UserModelAttribute)
public class RdfUserModelAttribute extends AbstractPersistable {
```

The constants used in the example are defined as: (the code is simplified)

```
class Foaf:
  String NAMESPACE = "http://xmlns.com/foaf/0.1/";
  String PREFIX = "foaf";
class Gomawe:
  String NAMESPACE = "http://fel.cvut.cz/gomawe/0.1/";
  String PREFIX = "gomawe";
  String UserModelAttribute = PREFIX + ":UserModelAttribute";
  String associatedUser = PREFIX + ":associatedUser";
```

To specify the property names, @RdfProperty annotation is used.

```
@ManyToOne
@RdfProperty(Gomawe.associatedUser)
private RdfUser associatedUser;
```

To query data, the JPA EntityManager can be used. Another possibility is to use more advanced parameterized query.

```
public T findById(RdfKey id) {
    return (T) getEntityManager().find(getPersistentClass(), id);
}

Query query = getEntityManager().createQuery(
    "where { ?result a " + getRdfsClass() + " . " +
            "?result " + RdfUserModelAttribute.Properties.ASSOCIATED_USER
             + " ??userParam . " +
    " }"
);
query.setParameter("userParam", user);
query.setHint(RdfQuery.HINT_ENTITY_CLASS, getPersistentClass());
return query.getResultList();
```

Achieved results

The expected result of this experiment is an advanced adaptive semantic learning web portal build on top of the ASF framework and demonstrating all its features. We have integrated all features that were evaluated in previous experiments into the project. We added support of ontologies and SPARQL queries over the data repository. We used adaptive algorithms included in the ASF.

5.7 Adaptive Shop

This experiment was designed by the author of this thesis as partial evaluation of the proposed AH theory. Implementation was realized by Petr Dobřička and was supervised by author of this work. The student will submit his thesis in January 2014.

To verify the general applicability of the model, and the generic ASF implementation, the details of Adaptive e-Shop use-case implementation will be explained. This project is the most-recent relational database ASF implementation. Moreover, Google Appengine Datastore has been used, that as a consequence led to several limitations, such as special primary key type, limited indexing and limited queries.

Compared to the ontology-based learning scenario, described in the previous section, the Adaptive e-Shop differs in two substantial principles. First, the storage data is currently not described by an ontology, and second, compared to the learning system, where adaptation is usually predefined by the author, the shop is an example of a data-model-driven AHS.

The absence of an ontology-based description can be a limiting factor for data integration scenario and for some ontology-based adaptation algorithms. However, the system architecture and user data classification is based on our model, and actually, most of the user-model-related implementation is already provided by the JPA module of the ASF framework.

The product screen of the application is depicted in Fig. 5.22. There are visible two adaptive links. One of them, the recommendation based on similar users is computed using the *user walk* algorithm included in the ASF framework, see Chapter 4. The second link is generated using a keyword-based match of product descriptions. The used algorithm is also included in the framework. The Table 5.4 summarizes the modeled attributes of the User Model, see Def. 3.8, Content Unit Model, see Def. 3.10, and User Profile, see Def. 3.7.

Figure 5.22: Adaptive shop prototype.

Table 5.4: Adaptive shop model attributes.

User Model		**Content Unit Model**		**User Profile**	
bought	number of bought items	sold	number of sold items	font	font size
rating	rating of a product	rating	average item rating	theme	selected theme

Achieved results

This experiment has proved the generality of the GOMAWE model in a non-typical application environment. We had to deal with several issues regarding the JPA implementation, and the JPA ASF module has been customized to support more variants of data-storage implementation. On the other hand, it was proven, that the ASF core layer is applicable without changes and satisfies all adaptive system requirements.

5.8 Summary

Our research was accompanied by series of projects that gave us partial results and evaluations.

Starting in the adaptive learning area, we have evaluated the benefits gained by using semantic data storage. We have successfully applied the same architecture in the area of on-line information systems and achieved data integration.

Experiments, that followed the initial prototype, were primarily focused on the adaptation process. Selected algorithms were implemented and a standalone program library was prepared to be used in future projects. The similar, separated approach was focused on the presentation layer. Our aim was to develop a program library of adaptive user interface components, that could be supported by lower layers of our framework and make the integration of adaptive behavior into web application easier.

A series of more and less successful projects were aimed on finding a suitable technological support for the development of adaptive hypermedia systems. Our attempts were confronted with outdated software platforms, low support on canceled projects and technology innovations. Our research resulted in the current architecture of ASF framework (Fig. 4.1).

In the final phase of our research, we have returned to the adaptive learning area. First phase of the implementation was based on well-known technologies – relational database and XML data exchange format. The implemented adaptive learning application was fully functional and was tested in real classes with students. In the second phase, the system was extended with ontologies and supports the RDF data format. We can state that the system is based on ASF framework, showing its benefits, and demonstrates the GOMAWE model design.

CHAPTER 6

Conclusion

In this last chapter of the thesis, we will summarize the achieved results of our work. After that, we will give an overview of additional, still unanswered questions, and possibilities for extensions in future research.

6.1 Contributions

The thesis has studied actual research problems in the areas of adaptive hypermedia modeling, personalization, adaptive semantic web and adaptive hypermedia systems engineering.

In the first chapter, we have asked several questions that motivated our work to improve adaptive hypermedia research area. In the following paragraphs, we will summarize our contributions, and we will answer the research questions.

Why is the generic ontology-based model necessary? Is it possible to devise the model from existing AHS?

The main reasons for the proposal of a new reference model were already mentioned as the motivation for our work. The GOMAWE model reflects new trends in personalization and adaptation. The model supports up-to-date user modeling styles and is generic enough to reflect emerging characteristics that may arise in the near future. A reference model provides a common terminology, allows comparing architecture and functionality of different systems and initiates the development of new systems based on the model.

To be able to devise the model, we have studied various methods and techniques used in adaptive hypermedia. Initial categorization as *adaptive presentation* and *adaptive navigation* has been shifted due to the technology advancements to allow a more granular division. The model components are tailored to fit various types of adaptive systems in regards of *content*, *presentation* and *navigation* adaptation support. Both the adaptation based on a pre-authored content and data-driven adaptation were considered. The model architecture is based on ontologies, because we have identified the interoperability issues in contemporary adaptive systems. The proposed ontology-based model provides means to solve these issues. However, in the matter of interoperability, additional research is required.

What are the elementary components of adaptive systems?

The answer to this question is provided within the main contributions of the thesis.

1. Generic ontology-based adaptive web system model was designed and described formally. Details are discussed in Chapter 3.

2. Multidimensional user model architecture supporting effective reasoning and data exchange was designed. For details see sections 3.3.2 and 3.3.3.

To be able to devise a generic ontology-based model for adaptive hypermedia, we had to ask the following subquestions:

- **What information about a user needs to be stored by the application to personalize the presentation?**
  We have reviewed, what user properties adaptive systems keep to perform the user modeling. We have identified an important classification of user properties – the *user profile* is used to store the preferences set explicitly by the user (adaptable system), and the *user model* is used to store data originating from the automatic user monitoring process (adaptive system). Additionally, user's context and goals can be optionally provided.

- **What information needs to be communicated while several adaptive systems cooperate and share user data?**
  In our work, we have focused on reusability and interoperability of user data in adaptive systems. Although, in some preliminary experiments, we have used

simple XML data exchange format, see Section 5.6.1, the true step to interoperability is the Semantic Web. We have verified, in series of experiments, how to store user model data in RDF and how the RDF data format can be used to exchange information about the user among separate systems. The main contribution is presented in Section 5.4.2, where also the possible application scenario is discussed.

- **What is the role of the Semantic Web in the area of AH?**
  Research on semantic web technologies has been a much sought-after topic in recent years. Semantic technologies provided advancements in a number of research fields. Merging the ideas of adaptive systems with semantic web, a new innovative field of Adaptive Semantic Web has emerged. In our work, we have identified the benefits of both technologies. An overview and comparison is provided in Chapter 2.9. The fusion of both technologies is an important aspect within our work. Some of the features of our model depend on the semantic representation. We have shown how ontologies help solving data integration issues and improve the user-data processing by reasoning. Reasoning helps in inferring the information not explicitly stored in the user model. Our research also focused on the implementation, and we have included selected technologies for accessing semantic triple stores into the ASF framework.

In the GOMAWE model design, we have identified and defined the basic components of an adaptive system. We have grouped the components into layers by their purpose and functionality. Architecture of adaptive systems consists of *storage, reasoning, integration* and *interface* layers. The details were presented in Chapter 3.

What methodologies should adaptive hypermedia developers follow?

The development process theory is aimed on hypermedia systems in general. Although the methodologies can be used for an adaptive application development too, they are missing important aspects of the adaptive process design and functionality. We have done a research and in Chapter 2.7, we summarized the methodologies applicable, or improved for adaptive hypermedia systems. The main achievements contributing to answer this question are:

1. Framework support for adaptive systems implementation in Java programming language. ASF provides implementation foundations and framework as a guidance for developers. For detail see Section 4.1.

2. Methodology for effective adaptive systems development. The extensions of the development process were presented in Section 4.2.

Subquestions outline the elementary steps towards the development methodology improvement.

- **What architecture is suitable and generally applicable for various user-adaptive systems?**
  To help developers to design and correctly implement adaptive hypermedia systems, we have defined a generally applicable architecture. The architecture is formally described as the GOMAWE model and implemented in ASF framework. The benefit that ASF can provide to developers is the framework defined by interfaces, easy extensibility and fundamental implementation based on selected technologies.

- **What best-practice solutions should we apply within the development of AHS?**
  Using best-practice solution is highly recommended in all kinds of software development. The ASF framework was designed following widely accepted design patterns. Details and relations were described in Chapter 4.

- **How can we evaluate the quality of a user adaptive system?**
  Evaluation is a crucial part of the adaptive system development process. Continuous evaluation of a developed system should become a common practice. In Chapter 2.8, we have summarized the most typical evaluation methods used for personalized applications. Based on the latest research, we propose a layered user-centric evaluation in the context of GOMAWE model.

6.2 Future work

The Adaptive Semantic Web is a complex area that emerged recently as a response to data-related issues in contemporary personalized applications. Our work aims to be a small step forward in the adaptive hypermedia research and there are many ways how our work can be extended and improved.

Research on ontologies is developing very fast, and semantic representation is becoming a key factor in modern applications. We have introduced a reference architecture model for personalized hypermedia systems, that utilize ontologies as semantic data representation. Subsequent research may focus in more detail on the **data modeling from the knowledge engineering perspective**. This is in compliance with ongoing research in this area [141].

In our work, we have mainly focused on the user side of the adaptive application. For a successful application, there needs to be a good and **advanced content management system for designing the adaptive behavior** and creating data annotations. Content management user interface is usually complex, and with additional requirements related to personalization specification, many usability problems arise. Personalized interface could help course designers, and it could promote a more intense use of adaptive systems in education.

In the future, it would be useful to **extend the implementation of ASF framework**. One of the most desired extensions would be the implementation of additional adaptive user interface components, that can be easily used by the web-application developer. Similar extension is possible in the reasoning layer. Several basic adaptive algorithms have been implemented, but many more can be added. Large possibilities of extension are in the scope of technologies. Our implementation is based on selected up-to-date technologies. These technologies can become outdated or simply unsuitable for a particular project. Due to the technology-independent core of the framework, the principles can be reused, and higher technology-dependent layers can be replaced. Even porting the framework to another programming language is possible.

In recent years, more attention is devoted to mobile devices. Such devices represent a suitable platform for utilization of a wide range of adaptation features. Context plays a very important role as a continually changing influence factor. **Application of adaptation in ubiquitous mobile environment** is a promising research field. Even if we were not focusing on mobile devices in our work, the GOMAWE model is compliant and a context-aware mobile application can be described in the terms of the model. The shift of personalized learning to mobile devices is the trend of the current research [144]. We plan to continue our work within this direction, conduct experiments in the field of mobile adaptive learning and extend the ASF framework to support the development for mobile platforms.

Extensions regarding semantic data are a topic for a long-term research. Many tools for data annotation, data extraction and for integration with social networking are needed. **The fusion of adaptive and semantic web is a promising trend and is heading towards a new perspective of hypermedia applications**.

Bibliography

[1] H. Abdi. Kendall rank correlation. In N. Salkind, editor, *Encyclopedia of Measurement and Statistics*. Sage Publications, Thousand Oaks (CA), 2007.

[2] P. Adamczyk, P. H. Smith, R. E. Johnson, and M. Hafiz. REST and Web Services: In Theory and in Practice. In E. Wilde and C. Pautasso, editors, *REST: From Research to Practice*, chapter 2, pages 35–57. Springer New York, New York, NY, 2011.

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

[4] M. Alotaibi and S. Bull. Combining Facebook and Open Learner Models to Encourage Collaborative Learning. In *ICCE Conference on Computer-supported Collaborative Learning (CSCL) and Learning Sciences*, Singapore, 2012.

[5] S. Amrouch and S. Mostefai. Ontology Interoperability Techniques, The State of the Art. *Journal of Information*, 2(1):20–27, 2012.

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

[7] O. Aubert and A. Beugnard. Adaptive Strategy Design Pattern. In *Proceedings of The Second Asian Pacific Pattern Languages of Programming Conference (KoalaPLoP 2001)*, pages 1–12, The Country Place, Melbourne, 2001.

[8] C. Bailey, W. Hall, D. E. Millard, and M. J. Weal. Adaptive hypermedia through contextualized open hypermedia structures. *ACM Transactions on Information Systems*, 25(4):16–es, Oct. 2007.

[9] C. P. Bailey. *An Agent-Based Framework to Support Adaptive Hypermedia.* PhD thesis, University of Southampton, 2002.

[10] M. Barla, P. Bartalos, M. Bieliková, R. Filkorn, and M. Tvarozek. Adaptive portal framework for Semantic Web applications. In S. Casteleyn, F. Daniel, P. Dolog, M. Matera, G.-J. Houben, and O. D. Troyer, editors, *Proceedings of the 2nd International Workshop on Adaptation and Evolution in Web Systems Engineering AEWSE 07*, Como, Italy, 2007. CEUR-WS.org.

[11] P. Barna, G.-j. Houben, and F. Frasincar. 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, 2004.

[12] D. Ben-Naim. *A Software Architecture that Promotes Pedagogical Ownership in Intelligent Tutoring Systems.* PhD thesis, University of New South Wales, Sydney, Australia, 2010.

[13] S. Berkovsky, D. Heckmann, and T. Kuflik. Addressing challenges of ubiquitous user modeling: Between mediation and semantic integration. In *Advances in Ubiquitous User Modelling*, pages 1–19. 2009.

[14] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286, Nov. 2007.

[15] T. Berners-Lee. What a semantic can represent, 1998.

[16] M. Bieliková and P. Nagy. Considering human memory aspects for adaptation and its realization in AHA! In W. Nejdl and K. Tochtermann, editors, *EC-TEL 2006: Technology Enhanced Learning*, pages 8–20. Springer-Verlag, 2006.

[17] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

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

[19] S. Boyce and C. Pahl. Developing Domain Ontologies for Course Content The Development of Ontologies. *Educational Technology & Society*, 10(3):275–288, 2007.

[20] D. Brickley and L. Miller. FOAF vocabulary specification 0.98, 2010.

[21] P. Brusilovsky. A tool for developing adaptive electronic textbooks on WWW. In *WebNet 96 Conference Proceedings*, 1996.

[22] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[23] P. Brusilovsky. Adaptive Navigation Support. *The Adaptive Web Methods and Strategies of Web Personalization*, 4321(4):263–290, 2007.

[24] P. Brusilovsky. Adaptive Hypermedia for Education and Training. In P. J. Durlach and A. M. Lesgold, editors, *Adaptive Technologies for Training and Education*, chapter 3, pages 46–66. Cambridge University Press, 2012.

[25] P. Brusilovsky and J. Eklund. Web-based Education for All : A Tool for Development Adaptive Courseware Web-based Education. In *Proceedings of Seventh International World Wide Web Conference*, volume 30, pages 291–300, 1998.

[26] P. Brusilovsky and N. Henze. Open Corpus Adaptive Educational Hypermedia. In P. Brusilovsky, A. Kobsa, and W. Neidl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 22, pages 671–696. Springer-Verlag, 2007.

[27] P. Brusilovsky and M. T. Maybury. From adaptive hypermedia to the adaptive web. *Communications of the ACM*, 45(5), 2002.

[28] P. Brusilovsky and E. Millán. User Models for Adaptive Hypermedia and Adaptive Educational Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 1, pages 3–53. 2007.

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

[30] S. Bui, I. Park, and W. Kettinger. The Mediating Role of Adaptive Personalization in Online Shopping. In *AMCIS 2012 Proceedings*, 2012.

[31] M. Bureš. *Formal Description of Adaptive Hypermedia System.* PhD thesis, Czech Technical University in Prague, 2006.

[32] M. Cannataro and A. Pugliese. XAHM: an XML-based Adaptive Hypermedia Model and its Implementation. In *Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 252–263. Springer, 2002.

[33] I. Cantador, A. Bellogín, and P. Castells. A multilayer ontology-based hybrid recommendation model. *AI Communications*, 2008.

[34] F. Carmagnola. Handling semantic heterogeneity in interoperable distributed user models. In *Advances in Ubiquitous User Modelling*, pages 20–36. 2009.

[35] F. Carmagnola and F. Cena. User identification for cross-system personalisation. *Information Sciences*, 179(1-2):16–32, Jan. 2009.

[36] F. Carmagnola, F. Cena, C. Gena, and I. Torre. A Semantic Framework for Adaptive web-based Systems. In *Proceedings of SWAP 2005, the 2nd Italian Semantic Web Workshop*, Trento, Italy, 2005.

[37] F. Carmagnola, F. Cena, C. Gena, and I. Torre. MUSE: A Multidimensional Semantic Environment for Adaptive Hypermedia Systems. In M. Bauer, B. Brandherm, J. Fürnkranz, G. Grieser, A. Hotho, A. Jedlitschka, and A. Kröner, editors, *Proceedings of Lernen, Wissensentdeckung und Adaptivität (LWA)*, pages 14–19, Saarbrücken, Germany, 2005. DFKI.

[38] B. Carminati, E. Ferrari, and A. Perego. Combining social networks and semantic web technologies for personalizing web access. In E. Bertino and J. B. D. Joshi, editors, *Collaborative Computing: Networking, Applications and Worksharing, 4th International Conference, CollaborateCom 2008*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pages 126–144. Springer, 2008.

[39] F. Cena and L. Aroyo. A semantics-based dialogue for interoperability of user-adaptive systems in a ubiquitous environment. In *User Modeling 2007*, pages 309–313, 2007.

[40] F. Cena and R. Furnari. A model for feature-based user model interoperability on the web. In *Advances in Ubiquitous User Modelling*, pages 37–54. 2009.

[41] F. Cena, S. Likavec, and F. Osborne. Property-based interest propagation in ontology-based user model. In *User Modeling, Adaptation, and Personalization*, pages 38–50, 2012.

[42] S. Ceri, P. Fraternali, A. Bongio, and P. Milano. 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, 2000.

[43] S. Cheng. *Rainbow: Cost-Effective Software Architecture-Based Self-Adaptation.* PhD thesis, Carnegie Mellon University, Pittsburgh, 2008.

[44] O. Conlan, V. Wade, C. Bruen, and M. Gargan. Multi-model , Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'02)*, pages 100–111. Springer, 2002.

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

[46] A. Cristea and L. Calvi. The three layers of adaptation granularity. In *Int. Conf. on User Modeling (UM2003)*, pages 4–14, Pittsburgh, US, 2003. Springer.

[47] A. I. Cristea. What can the Semantic Web do for Adaptive Educational Hypermedia ? E-Learning , Adaptive Hypermedia and the Semantic Web. 7:40–58, 2004.

[48] A. I. Cristea and A. de Mooij. LAOS: Layered WWW AHS authoring model and their corresponding algebraic operators. In *Int. World Wide Web Conference (WWW'03)*, Budapest, Hungary, 2003.

[49] E. Damiani and S. David. Open World Reasoning in Semantics-Aware Access Control: a Preliminary Study. In *Proceedings of SWAP 2005, the 2nd Italian Semantic Web Workshop*, pages 1–15, 2005.

[50] J. Day. The (un)revised OSI reference model. *ACM SIGCOMM Computer Communication Review*, 25(5):39–55, 1995.

[51] P. S. de Assis and D. Schwabe. A General Meta-model for Adaptive Hypermedia Systems. In P. de Bra and W. Nejdl, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2004*, Lecture Notes in Computer Science, pages 433–436. Springer, 2004.

[52] P. S. de Assis and D. Schwabe. A Semantic Meta-model for Adaptive Hypermedia Systems. In P. de Bra and W. Nejdl, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2004*, volume 3137 of *Lecture Notes in Computer Science*, pages 360–365, Eindhoven, The Netherlands, 2004. Springer.

[53] P. De Bra, A. Aerts, B. Berden, B. D. Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash. AHA! The Adaptive Hypermedia Architecture. In *Proc. of the fourteenth ACM conference on Hypertext and Hypermedia*, volume 4, pages 81–84. ACM Press, 2003.

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

[55] P. De Bra, D. Smits, K. V. D. Sluijs, A. I. Cristea, and M. Hendrix. GRAPPLE: Personalization and adaptation in learning management systems. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2010)*, pages 3029–3038, Toronto, Canada, 2010.

[56] J. de Bruijn, F. Martín-Recuerda, M. Ehrig, A. Polleres, and L. Predoiu. Ontology Mediation Management v1. Deliverable D4.4.1. Technical report, SEKT, 2005.

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

[58] P. de Vrieze. *Fundaments of Adaptive Personalisation*. PhD thesis, Radboud University Nijmegen, 2006.

[59] P. de Vrieze, P. van Bommel, J. Klok, and T. van der Weide. Towards a Two-Dimensional Framework for User Models. In *Proceedings of the MAWIS03 workshop attached to the OOIS03 conference*, volume 1, Geneva, 2003.

[60] P. de Vrieze, P. Van Bommel, and T. Van der Weide. A Generic Adaptivity Model in Adaptive Hypermedia. In *Proceedings of the adaptive hypermedia conference*, Lecture Notes in Computer Science, pages 344–347. Springer, 2004.

[61] P. de Vrieze and T. P. van der Weide. GAM : A Generic Model for Adaptive Personalisation. Technical report, Radboud University Nijmegen, 2006.

[62] P. Dolog, N. Henze, W. Nejdl, and M. Sintek. The personal reader: Personalizing and enriching learning resources using semantic web technologies. In P. De Bra and W. Nejdl, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, Third International Conference, AH 2004*, Lecture Notes in Computer Science, pages 85–94. Springer, 2004.

[63] P. Dolog and M. Schäfer. A framework for browsing, manipulating and maintaining interoperable learner profiles. In *User modeling 2005*, Lecture Notes in Computer Science, pages 397–401. Springer, 2005.

[64] C. Dorn, D. Schall, and S. Dustdar. Context-aware adaptive service mashups. In M. Kirchberg, P. C. K. Hung, B. Carminati, C.-H. Chi, R. Kanagasabai, E. D. Valle, K.-C. Lan, and L.-J. Chen, editors, *2009 IEEE Asia-Pacific Services Computing Conference (APSCC)*, number c, pages 301–306. Ieee, Dec. 2009.

[65] E. Dumbill. Building the Semantic Web, 2001.

[66] S. Federici. Usability evaluation : models, methods, and applications. In *International Encyclopedia of Rehabilitation*. 2010.

[67] D. Fensel. *Ontologies:: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2000.

[68] R. T. Fielding and R. N. Taylor. Principled design of the modern Web architecture. *ACM Transactions on Internet Technology*, 2(2):115–150, May 2002.

[69] J. Fink, A. Kobsa, and J. Schreck. Personalized Hypermedia Information Provision through Adaptive and Adaptable System Features : User Modelling , Privacy and Security Issues. In A. P. Mullery, M. Besson, M. Campolargo, R. Gobbi, and R. Reed, editors, *Intelligence and Services Networks: Technology for Cooperative Competition*, Lecture Notes in Computer Science, pages 459–467. Springer, 1997.

[70] T. Fredrich. RESTful Service Best Practices. 2012.

[71] C. Fröschl. User modeling and user profiling in adaptive e-learning systems. Master's thesis, Graz University of Technology, 2005.

[72] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Professional Computing Series. Addison-Wesley Longman Publishing Co., Inc., 1995.

[73] V. M. García-barrios, F. Mödritscher, and C. Gütl. Personalisation versus Adaptation? A User-centred Model Approach and its Application. In *International Conference on Knowledge Management (I-KNOW)*, number Iicm, pages 120–127, Graz, 2005.

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

[75] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. In P. Brusilovsky, A. Kobsa, and W. Neidl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 2, pages 54–89. Springer, Berlin, 2007.

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

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

[78] F. Ghali and A. I. Cristea. Social reference model for adaptive web learning. In *Advances in Web Based LearningICWL 2009*, pages 162 – 171, Berlin, Heidelberg, 2009. Springer-Verlag.

[79] D. Giordano and I. Kavasidis. A semantic-based and adaptive architecture for automatic multimedia retrieval composition. In *Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*, pages 181–186, 2011.

[80] I. Gorton, Y. Liu, and N. Trivedi. An extensible and lightweight architecture for adaptive server. *Softw., Pract. Exper.*, 38(8):853–883, 2008.

[81] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2005.

[82] M. Grove. Empire RDF & SPARQL Meet JPA, 2010.

[83] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[84] M. Grčar. User profiling: Collaborative filtering. In *proceedings of SiKDD 2004*, Ljubljana, Slovenia, 2004.

[85] M. Grčar, D. Mladenič, and M. Grobelnik. Data quality issues in collaborative filtering. In *the proceedings of ESWC05*, number Section 8, Heraklion, Greece, 2005.

[86] D. Guinard, I. Ion, and S. Mayer. In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective. In A. Puiatti and T. Gu, editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services (MobiQuitous 2011)*, pages 326–337, Copenhagen, Denmark, 2011. Springer.

[87] F. Halasz and M. Schwartz. The Dexter Hypertext. *Communications of the ACM*, 37(2):30–39, 1994.

[88] J. Hannon, E. Knutov, P. De Bra, M. Pechenizkiy, K. McCarthy, and B. Smyth. Bridging Recommendation and Adaptation: Generic Adaptation Framework-Twittomender compliance case-study. In *Proceedings of 2nd DAH'2011 Workshop on Dynamic and Adaptive Hypertext*, number march, pages 1–9, 2011.

[89] D. Heckmann, T. Schwartz, B. Brandherm, and M. Schmitz. GUMO The General User Model Ontology. pages 428–432, 2005.

[90] S. R. Heiyanthuduwage and D. D. Karunaratne. A learner oriented ontology of metadata to improve effectiveness of learning management systems. In *Third International Conference on eLearning for Knowledge-Based Society*, pages 1–6, Bangkok, Thailand, 2006.

[91] N. Henze. *Adaptive hyperbooks: Adaptation for project-based learning resources.* PhD thesis, Universiat Hannover, 2000.

[92] N. Henze and W. Nejdl. Adaptivity in the KBS hyperbook system. In *2nd Workshop on Adaptive Systems and User Modeling on the WWW*, Toronto, Canada, 1999.

[93] S. Y. Ho. Opportunities and challenges of mobile personalization: An exploratory study. In S. Newell, E. A. Whitley, N. Pouloudi, J. Wareham, and L. Mathiassen, editors, *ICIS 17th European Conference on Information Systems*, pages 1211–1222, 2009.

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

[95] W. Huang, L. Zhang, J. Zhang, and M. Zhu. Semantic Focused Crawling for Retrieving E-Commerce Information. *Journal of Software*, 4(5):436–443, July 2009.

[96] R. Hübscher and S. Puntambekar. Modeling learners as individuals and as groups. In P. De Bra and W. Nejdl, editors, *Adaptive Hypermedia 2004 Conference Proceedings*, pages 300–303. Springer, 2004.

[97] M. Huhns and M. Singh. Ontologies for agents. *IEEE Internet Computing*, 1(6):81–83, 1997.

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

[99] L. Kalinichenko and M. Missikoff. Ontological modeling. In *Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003*, 2003.

[100] J. Kay, B. Kummerfeld, and P. Lauder. Personis: A server for user models. In P. De Bra, P. Brusilovsky, and R. Conejo, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems (AH2002)*, pages 203–212, Malaga, Spain, 2002. Springer.

[101] E. F. Kendall and M. E. Dutra. An Introduction and UML Profile for the Web Ontology Language (OWL). In *Fifth International Conference on the Unified Modeling Language (¡¡UML¿¿ 2002)*, number 408, pages 1–39, Dresden, Germany, 2002.

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

[103] E. Knutov, P. De Bra, and M. Pechenizkiy. Provenance meets adaptive hypermedia. *Proceedings of the 21st ACM conference on Hypertext and hypermedia HT 10*, page 93, 2010.

[104] E. Knutov, P. De Bra, and M. Pechenizkiy. Generic Adaptation Framework: A Process-Oriented Perspective. *Journal Of Digital Information*, 12(1), 2011.

[105] A. Kobsa, J. Koenemann, and W. Pohl. Personalised hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(02):111–155, Dec. 2001.

[106] A. Kobsa and M. Teltzrow. Convincing Users to Disclose Personal Data. *Privacy-Enhanced Personalization*, pages 19–21, 2006.

[107] N. Koch. *Software Engineering for Adaptive Hypermedia Systems and Development Process*. PhD thesis, 2000.

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

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

[110] P. Křemen and Z. Kouba. Ontology-Driven Information System Design. *IEEE Transactions on Systems, and Cybernetics*, 42(3):334–344, 2012.

[111] D. Lange. 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, 1994. IEEE Computer Society Press.

[112] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[113] B. Magoutas and G. Mentzas. SALT: A semantic adaptive framework for monitoring citizen satisfaction from e-government services. *Expert Systems with Applications*, 37(6):4292–4300, June 2010.

[114] M. D. L. Martinez-Villaseñor, M. Gonzalez-Mendoza, and N. Hernandez-Gress. Towards a ubiquitous user model for profile sharing and reuse. *Sensors (Basel, Switzerland)*, 12(10):13249–83, Jan. 2012.

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

[116] S. Middleton, D. Roure, and N. Shadbolt. Ontology-based recommender systems. In S. Staab and R. Studer, editors, *Handbook on ontologies*, pages 779–796. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[117] M. Miettinen, J. Kurhila, and H. Tirri. On the prospects of intelligent collaborative e-learning systems. In *Proc. 12th International Conference on Artificial Intelligence in Education (AI-ED 2005)*, pages 483–490. IOS Press, 2005.

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

[119] D. Moore. *The basic practice of statistics.* WH Freeman Company, 1996.

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

[121] J. L. Myers and A. D. Well. *Research design and statistical analysis.* 2003.

[122] W. Nejdl and B. Wolf. EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network. In M. Frank, N. F. Noy, and S. Staab, editors, *Semantic Web Workshop, WWW02 Conference, Hawai, USA*, CEUR Workshop Proceedings. CEUR-WS.org, 2002.

[123] G. Ng. Open vs Closed world, Rules vs Queries: Use cases from Industry. In B. C. Grau, I. Horrocks, B. Parsia, and P. F. Patel-Schneider, editors, *OWL: Experiences and Directions (OWLED 2005)*, CEUR Workshop Proceedings. CEUR-WS.org, 2005.

[124] M. Niederhausen, S. Karol, U. Aß mann, and K. Meiß ner. HyperAdapt: Enabling Aspects for XML. In M. Gaedke, M. Grossniklaus, and O. Díaz, editors, *Web Engineering, 9th International Conference, ICWE 2009*, Lecture Notes in Computer Science, pages 461–464, San Sebastián, 2009. Springer.

[125] J. Nielsen. *Usability Engineering.* Academic Press, Boston, MA, 1993.

[126] O. Nouali and A. Belloui. Using semantic web to reduce the cold-start problems in recommendation systems. *2009 Second International Conference on the Applications of Digital Information and Web Technologies*, pages 525–530, Aug. 2009.

[127] N. F. Noy and M. A. Musen. SMART : Automated Support for Ontology Merging and Alignment 1 . Merging Versus Alignment. pages 1–24, 1999.

[128] N. F. Noy and M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 450–455, 2000.

[129] D. Oberle, R. Volz, S. Staab, and B. Motik. An extensible ontology software environment. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, chapter III., pages 299–320. Springer, 2004.

[130] J. F. Ohene-Djan. *A Formal Approach to Personalisable, Adaptive Hyperlink-Based Systems*. PhD thesis, Goldsmiths College University of London, 2002.

[131] D. O'Sullivan, D. C. Wilson, and B. Smyth. Preserving recommender accuracy and diversity in sparse datasets. In I. Russell and S. M. Haller, editors, *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, pages 139–143. AAAI Press, 2003.

[132] E. Pampalk, T. Pohle, and G. Widmer. Dynamic playlist generation based on skipping behavior. In *Proc. of the 6th ISMIR Conference*, number C, pages 634–637, London, UK, 2005.

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

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

[135] D. Parsons and S. Schroeder. Adaptive Information Systems for the Web 2.0: Developing a Mobile Learning Architecture. In *Proceedings of the UK Academy for Information Systems Conference*, pages 1–10, 2006.

[136] C. Pautasso and E. Wilde. Why is the web loosely coupled?: a multi-faceted metric for service design. In *Proc. of the 18th international conference on World Wide Web (WWW '09)*, pages 911–920, Madrid, Spain, 2009. ACM.

[137] C. Pautasso, O. Zimmermann, and F. Leymann. RESTful web services vs. "big" web services: making the right architectural decision. In *WWW 08: Proceeding of the 17th international conference on World Wide Web*, pages 805–814, New York, NY, USA, 2008. ACM.

[138] S. Piao, D. Bental, and J. Whittle. A Pilot Study: Deriving a Users' Goal Framework from a Corpus of Interviews and Diaries. In *SemRel II-Enhancing Resources and Applications*, 2012.

[139] N. Primus. *A generic framework for evaluating Adaptive Educational Hypermedia authoring systems.* Master theses, University of Twente, 2005.

[140] S. Raunich and E. Rahm. ATOM: Automatic target-driven ontology merging. *2011 IEEE 27th International Conference on Data Engineering*, pages 1276–1279, Apr. 2011.

[141] B. Ravishankar. Web Personalization Using Ontology: A Survey. *IOSR Journal of Computer Engineering*, 1(3):37–45, 2012.

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

[143] G. Salton and M. J. McGill. *Introduction to modern information retrieval.* McGraw Hill Book Co., New York, 1983.

[144] D. G. Sampson and P. Zervas. Context-Aware Adaptive and Personalized Mobile Learning. In *3rd International Conference for e-learning and Distance Education*, Riyadh, 2013.

[145] J. Schreck. *Security and privacy in user modeling.* PhD thesis, Universität Gesamthochschule Essen, 2001.

[146] G. Shakhnarovich, T. Darrell, and P. Indyk. *Nearest-neighbor methods in learning and vision: theory and practice.* The MIT Press, 2006.

[147] I. Singh, B. Stearns, and M. Johnson. *Designing Enterprise Applications with the J2EE Platform.* Addison Wesley, second edition, 2002.

[148] K. Siorpaes and M. Hepp. Ontogame: Weaving the semantic web by online games. In *The Semantic Web: Research and Applications*, pages 1–15, 2008.

[149] D. Sleeman. UMFE: A user modelling front-end subsystem. *International Journal of Man-Machine Studies*, 23(1):71–88, 1985.

[150] S. J. Soltysiak and I. B. Crabtree. Automatic learning of user profiles towards the personalisation of agent services. *BT Technology Journal*, 16(3):110–117, 1998.

[151] S. Sosnovsky, P. Brusilovsky, M. Yudelson, A. Mitrovic, M. Mathews, and A. Kumar. Semantic Integration of Adaptive Educational Systems. In T. Kuflik, S. Berkovsky, F. Carmagnola, D. Heckmann, and A. Krüger, editors, *Advances in Ubiquitous User Modelling*, pages 134–158. Springer, 2009.

[152] J. Sowa. *Knowledge representation: logical, philosophical, and computational foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[153] B. Steichen, H. Ashman, and V. Wade. A comparative survey of Personalised Information Retrieval and Adaptive Hypermedia techniques. *Information Processing & Management*, 48(4):698–724, 2012.

[154] L. Stojanovic, S. Staab, and R. Studer. eLearning based on the Semantic Web. In *WebNet2001 - World Conference on the WWW and Internet*, pages 23–27, Orlando, Florida, USA, 2001.

[155] L. Stojanovic, N. Stojanovic, and S. Handschuh. Evolution of the metadata in the ontology-based knowledge management systems. In *1st German Workshop on Experience Management: Sharing Experiences about the Sharing of Experience*, Berlin, 2002.

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

[157] K. van der Sluijs and G. Houben. Towards a generic user model component. In *Proc. of PerSWeb05, Workshop on Personalisation on the Semantic Web*, pages 47–56, Edinburgh, UK, 2005.

[158] J. D. Velásquez and V. Palade. *Adaptive web Sites: A knowledge extraction from web data approach*. 2008.

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

[160] M. Šimko, M. Barla, and M. Bieliková. ALEF : A Framework for Adaptive Web-Based Learning 2.0. *Ifip International Federation For Information Processing*, 324:367–378, 2010.

[161] Y. Wang and A. Kobsa. Technical solutions for privacy-enhanced personalization. *Intelligent User Interfaces: Adaptation and Personalization Systems and Technologies*, pages 326–353, 2009.

[162] G. Weber, H. Kuhl, and S. Weibelzahl. Developing adaptive internet based courses with the authoring system NetCoach. In S. Reich, M. Tzagarakis, and P. De Bra, editors, *Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 226–238. Springer, 2002.

[163] G. Weber and M. Specht. User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. In A. Jameson, C. Paris, and C. Tasso, editors, *Proceedings of 6th International Conference on User Modeling*, pages 289–300. Springer, 1997.

[164] S. Weibelzahl. *Evaluation of Adaptive Systems.* PhD thesis, University of Trier, 2002.

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

[166] H. Wu, G. Houben, and P. D. Bra. AHAM: a reference model to support adaptive hypermedia authoring. In *InfWet'98 Conference*, pages 77–88, 1998.

Works supervised by the author and related to the thesis

[S.1] Z. Gerlický. Adaptive web portal. Master Thesis, Czech Technical University in Prague, 2008. (In Czech)

[S.2] M. Stružský. Collaborative filtering for adaptive web. Bachelor Thesis, Czech Technical University in Prague, 2009. (In Czech)

[S.3] V. Pešek. Adaptive e-learning system – user interface. Bachelor Thesis, Czech Technical University in Prague, 2009. (In Czech)

[S.4] M. Matuška. Adaptive e-learning system – authoring interface. Bachelor Thesis, Czech Technical University in Prague, 2009. (In Czech)

[S.5] J. Hlinka. Adaptive algorithms. Bachelor Thesis, Czech Technical University in Prague, 2010. (In Czech)

[S.6] L. Koranda. A Web service providing semantic data. Bachelor Thesis, Czech Technical University in Prague, 2010. (In Czech)

[S.7] M. Snížek. Adaptive e-learning system. Bachelor Thesis, Czech Technical University in Prague, 2011. (In Czech)

[S.8] O. Harcuba. Graphics environment for adaptive web applications. Bachelor Thesis, Czech Technical University in Prague, 2012. (In Czech)

[S.9] M. Hořejší. Adaptive Java course. Master Thesis, Czech Technical University in Prague, 2012. (In Czech)

Refereed publications of the author

[A.1] M. Balík., I. Jelínek. Modelling of Adaptive Hypermedia Systems. In *Proceedings of the Interenational Conference on Computer Systems and Technologies (CompSysTech06)*, pages V-8-1-V-8-6, Varna, 2006. Academy of Sciences.

M. Balík., I. Jelínek. Modelling of Adaptive Hypermedia Systems. *The Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology*, Vol 23, N 1-4, p 99-104, 2006.

[A.2] M. Balík., I. Jelínek. General Architecture of Adaptive and Adaptable Information Systems. In *ICEIS Doctoral Consortium*, pages 29-34, Funchal, Madeira-Portugal, 2007.

[A.3] M. Balík., I. Jelínek. Application of Adaptive Hypermedia Model in e-Learning. In *Proceedings of e-Learning Conference '07*, pages 60-64, Rousse, 2007. Bulgarian Chapter of ACM.

[A.4] M. Balík., I. Jelínek. Improving the Adaptive Systems Using Ontologies and Semantic Web Technologies. In *4th International Conference on Innovations in Information Technology, IIT'07*, pages 738-740, Dubai, 2007. UAE University.

[A.5] M. Balík., I. Jelínek. Towards Semantic Web-based Adaptive Hypermedia Model. In *ESWC Ph.D. Symposium*, pages 1-5, Tenerife, Spain, 2008.

[A.6] M. Balík., I. Jelínek. Towards Data Interoperability in Adaptive Environments. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pages 2480-2485, Las Vegas, USA, 2008.

[A.7] M. Balík., I. Jelínek. Experimental Adaptive Web Portal with Semantic Data Store. In *Third International Workshop on Semantic Media Adaptation and Personalization*, pages 189-192, Prague, Czech Republic, 2008. Ieee.

[A.8] M. Balík., I. Jelínek. Collaborative Filtering Support for Adaptive Hypermedia. In *ICWE 2009 Doctoral Consortium Proceedings*, pages 55-60, San Sebastian, Spain, 2009. University of the Basque Country.

[A.9] M. Balík., I. Jelínek. Development of an Adaptive Learning Environment. *International Journal on Information Technologies and Security*, IV(2):19-32, 2012.

[A.10] M. Balík., I. Jelínek. Adaptive System Framework: A Way to a Simple Development of Adaptive Hypermedia Systems. In *ADAPTIVE 2013, The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications*, pages 20-25, Valencia, Spain, 2013. IARIA.

[A.11] M. Balík., I. Jelínek. Generic Ontology-based Model for Adaptive Web Environments: A Revised Formal Description Explained within the Context of its Implementation. In *CIT 2013, The 13th IEEE International Conference on Computer and Information Technology* (in press).

Unrefereed publications of the author

[A.12] M. Balík., I. Jelínek. User Modelling in Adaptive Systems. In *Proceedings of WORKSHOP 2007*, Prague, CTU, 2007.

[A.13] I. Jelínek, L. Hapalová, T. Havryluk, T. Kadlec, M. Balík, R. Jun, M. Švihla, M. Bureš, M. Ota. Inteligentní webové technologie v e-learningu. In *Technologie pro e-vzdělávání TPEV 2007*, pages 23-30, Prague, Czech Republic, 2007. (In Czech)

[A.14] M. Balík., I. Jelínek. A General Model for Personalized Web Design. In *Proceedings of WORKSHOP 2008*, Prague, CTU, 2008.

[A.15] M. Balík. Ontological Model for Personalized Web Design. Dissertation Thesis Proposal DCSE-DTP-2008-03, Department of Computer Science and Engineering, Czech Technical University, Prague, Jan. 2008.

[A.16] M. Balík., I. Jelínek. Adaptive Hypermedia with Semantic Data Store. In *Proceedings of WORKSHOP 2009*, Prague, CTU, 2009.

[A.17] M. Balík., I. Jelínek. Model for Adaptive Environments. In *Proceedings of WORKSHOP 2010*, Prague, CTU, 2010.

Citations

- Paper [A.1] has been cited in:

  1. A. Memari, J.M. Gomez. A Model for adaptive applications on the Semantic Web. In *Information and Communication Technologies: From Theory to Applications*, pages 1–6, 2008.
  2. A. Paramythis. Adaptive Systems: Development, Evaluation and Evolution. *PhD Thesis*, Johannes Kepler University Linz, 2009.
  3. B. Jankowska. Architectural Frameworks for Automated Content Adaptation to Mobile Devices Based on Open-Source Technologies. *PhD Thesis*, Faculty of Economics – European University Viadrina, 2007.

- Paper [A.2] has been cited in:

  1. E. Knutov, P. De Bra, M. Pechenizkiy. AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, 15 (1): 5–38, ISSN 1361–4568, 2009.
  2. E. Knutov. Generic Adaptation Framework for unifying adaptive web-based systems. *PhD Thesis*, Eindhoven University of Technology, 2012.
  3. J. Cadavid. Reference Model for Adaptive and Intelligent Educational Systems supported by Learning Objects. *PhD Thesis*, Universidad Nacional de Colombia, 2012.
  4. AHHM Mohamed. ARTS: An Adaptive Recursive Tutoring System. In *Proceedings of the 2011 Developments in E-systems Engineering*, ISBN 978–0–7695–4593–6, pages 255–260, 2011. IEEE Computer Society.
  5. AHHM Mohamed. A Framework for a Web-based Adaptive Tutoring System (FWATS). *PRIME journal*, ISSN 1744–2494, 2010.

- Paper [A.5] has been cited in:

    1. T. Vengattarman, P. Dhavachelvan. An agent-based personalized E-learning environment: Effort prediction perspective. In *Proceedings of the IEEE International Conference on Intelligent Agent & Multi-Agent Systems*, 2009.
    2. B. Raufi, J. Georgieva, A. Luma, F. Ismaili. An adaptation algorithm for adaptive web based systems based on link structure and document similarity. In *Proceedings of the 9th WSEAS International Conference on Telecommunications and Informatics*, pages 29–34, ISSN 1790–5117, 2010.
    3. B. Raufi, J. Georgieva. Adaptive web-based systems: from framework to implementation. In *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*, pages 521–527, 2010.
    4. E. Knutov, P. De Bra, M. Pechenizkiy. Generic adaptation framework: a process-oriented perspective. *Journal of Digital Information*, 12 (1), ISSN 1368–7506, 2011.
    5. S. Misó, M. Hornos, M. Rodríguez. In *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 107-111, ISBN 978-1-4577-1676-8, 2011.
    6. E. Knutov. Generic Adaptation Framework for unifying adaptive web-based systems. *PhD Thesis*, Eindhoven University of Technology, 2012.
    7. P. De Bra, E. Knutov, D. Smits, N. Stash, V. F. C. Ramos. GALE: a generic open source extensible adaptation engine. *New Review of Hypermedia and Multimedia*, 19 (2), pages 182–212, 2013.

- Paper [A.6] has been cited in:

    1. A. Paramythis. Adaptive Systems: Development, Evaluation and Evolution. *PhD Thesis*, Johannes Kepler University Linz, 2009.

APPENDIX A

Lists of abbreviations

AH Adaptive Hypermedia

AHAM Adaptive Hypermedia Application Model

AHS Adaptive Hypermedia System

ASF Adaptive System Framework

DTD Document Type Definition

FOAF Friend of a Friend

FOHM The Fundamental Open Hypermedia Model

GAF Generic Adaptation Framework

GAHM The Goldsmiths Adaptive Hypermedia Model

GAM Generic Adaptivity Model

GAP Generic Adaptation Process

HCI Human Computer Interaction

IR Information Retrieval

JPA Java Persistence API

JSF Java Server Faces

JSF JavaServer Faces

LAOS Layered WWW AHS Authoring Model

LMS Learning Management System

LO Learning Object

LOM Learning Object Metadata

MVC Model-View-Controller

NS Namespace

OWL Web Ontology Language

RDF Resource Description Framework

SCORM Shareable Content Object Reference Model

SOAP Simple Object Access Protocol

SPARQL SPARQL Protocol and RDF Query Language

UI User Interface

UML Unified Modeling Language

URI Uniform Resource Identifier

XAHM XML Adaptive Hypermedia Model

XML Extensible Markup Language