Sharing User Models for Adaptive Hypermedia Applications

Mária Bieliková and Jaroslav Kuruc
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
{bielik,kuruc}@fiit.stuba.sk

Abstract

In this paper we present an approach to sharing a user model among several adaptive hypermedia applications. Current adaptive hypermedia applications often realize the user model as the internal part of the system without any possibility to share this model with other applications. We introduce a web service called UMoWS *(User Model Web Service). The web service acts as a store of user characteristics represented in pre-defined ontologies that define semantics of the stored knowledge. Access and corresponding privileges to the web service is managed by the user through the management interface. The user is also allowed to inspect/modify the state of the user model.*

1. Introduction

The aim of adaptive hypermedia applications is to reflect particular features of a user or an environment in order to adapt presented content, its layout or navigation within the information space. User characteristics are defined in the user model, which together with the domain model and the adaptation model constitute basic components of the adaptive hypermedia application [4, 9]. The role of the user model is to represent various user's characteristics such as user's knowledge, preferences, interests, tasks or goals. It is considered as the basic source of the adaptation used in the adaptation model when adapting content represented in the domain model.

The user model is usually managed by the adaptive hypermedia application itself. It is realized as an internal part of the adaptive hypermedia system, without any possibility to share this model with other adaptive hypermedia applications. This approach poses several problems mainly related to the necessity of the user model initialization separately for each adaptive hypermedia application, lack of semantics in user characteristics specification, or non-transparent content of the user model, which can cause the user mistrust.

Regarding to the mentioned problems, an effort to separate the user model from the adaptive hypermedia application architecture is the meaningful way towards the cross-application adaptation. We propose in this paper a solution based on definition of the web service acting as a store of user characteristics for purpose of adaptation within various web-based hypermedia applications.

Other problem with actual user models is related to the fact that the user does not have the control nor the information on what is stored in the user model. An adaptive hypermedia application can build (a part) of the user model automatically by collecting data about the user regarding his interaction with the system. This information is used later for the adaptation purposes. Collected data are often hidden to the user, what may cause in a fear of misuse of this information. The scepticism may lead to refusing of usage of such systems.

A contribution to the solution of the problem of trust is a transparent and comprehensible presentation of the user model state to the user with a possibility to affect it. Only few existing adaptive hypermedia applications support a mechanism for inspecting/changing the state of the user model by the user itself, usually through a form based wizards. Therefore, a uniform way of inspecting/changing the user model independent on the features of the adaptive hypermedia application is needed. Our proposal presented in this paper includes also solution to this issue.

The paper is structured as follows. In Section 2 we characterize the representation of the user model in current adaptive hypermedia applications, together with the mechanism for updating the user model and its utilization for the adaptation purposes. Next, in Section 3, we describe proposed architecture for sharing a user model among several adaptive hypermedia applications. Section 4 discusses related works. Finally, we give conclusions.

2. Separate User Model

The source of the adaptation in an adaptive hypermedia application is the user model, which represents user characteristics significant from the adaptation point of view. This could be the user's knowledge of the domain in educational adaptive hypermedia application, or user's preferences, interests, goals, etc.

From the modeling point of view, user characteristics are usually represented as *attribute-value* pairs. We distinguish *domain dependent* attributes and *domain independent* attributes. Domain dependent attributes extend definition of concepts in the domain model to store user specific values for these concepts. Well-known example in the educational hypermedia is an attribute representing the level of user knowledge related to a particular concept. Attributes independent on the domain represent characteristics useful without consideration the specific domain, e.g. user-preferred language, media or color scheme of the presentation.

One example of representing domain dependent characteristics as an extension of the domain model, where the definition of the concepts is expanded with the definition of the domain dependent attributes is the system AHA! [3]. The definition of the user model lies in defining unique attributes for individual concepts. Attributes can be of type boolean, integer or string. For every user, an instance of the domain model is maintained and used for adaptation purposes.

From the user model maintenance point of view, following aspects should be covered when working with the user model (for both domain dependent and domain independent attributes):

- *initialization* of the user model,
- user *observation* and collecting system usage data,
- processing of the observed information by *updating* the user model,
- *usage* of the user model for the purposes of the adaptation.

Initialization of the user model might be realized by different techniques. For example, the definition of the attribute in the AHA! system contains except the attribute identifier and value type also a default value. Thus, the presentation is the same for all users who are working with the system for the first time. Another solution is to classify users into several groups and assign default values for these groups [1]. Some educational adaptive hypermedia applications require the user to go through an initial test to observe the user knowledge and initialize the user model appropriately.

User observation consists of monitoring the user activity during the usage of the adaptive hypermedia application. The system monitors links the user is following, items she is selecting from the application menu, time spent on particular pages, etc. Some adaptive hypermedia applications prompt the user to enter additional information related to the behavior during the system usage. For example, educational adaptive hypermedia applications sometimes provide the user with the possibility to enter innate sight of understanding presented information.

Collected data serve as a source for adaptation defined by the adaptation model. Adaptation model defines knowledge on reflecting observed data about the user together with the method of influencing the adaptation by the user model state. Often adaptation knowledge are represented using rules. In the AHA! system, the definition of a concept contains rules evaluated when the page containing information fragments corresponding with the particular concept is showed. The rules determine which attribute values are changed, where new value is often a result of an expression.

The code below shows an example of the usage of the user model for the purposes of adaptation. A sample fragment of domain/user model definition in the AHA! system is shown. We present an example related to the domain of learning programming inspired by [10].

```
<concept>
  <name>recursion</name>
  <resource>recursion.xhtml</resource>
  <req>lisp.interest &gt; 50</req>

  <attribute name="access" type="bool">
    <default>false</default>
    <generateListItem isPropagating="true">
      <req>personal.learningStrategy ==
           "TopDown"</req>
      <trueActions>
        <action>
          <concept>schema-1</concept>
          <attr>suitability</attr>
          <expr>schema-1.suitability +
                50</expr>
        </action>
      </trueActions>
    </generateListItem>
    <generateListItem isPropagating="true">
      <req>personal.learningStrategy ==
           "BottomUp"</req>
      <trueActions>
        <action>
          <concept>example-1</concept>
          <attr>suitability</attr>
          <expr>example-1.suitability +
                50</expr>
        </action>
      </trueActions>
    </generateListItem>
  </attribute>
</concept>
```

Given code shows a definition of the `recursion` concept representing an information fragment about the recursion algorithm in the field of functional programming (using LISP language). The content of the information fragment is defined in the `recursion.xhtml` page. The concept is considered to be suitable when the expression `lisp.interest > 50` is fulfilled (which means that the user is interested in LISP programming). Thus, the current value stored in the user model for the attribute `interest` defined in the `lisp` concept is considered. Since the value differs for particular users, the concept may be more or less suitable.

Definition of a concept includes also the definition of the `access` attribute, which has associated rules (a set of `generateListItem` definitions) for updating the user model. When the `recursion.xhtml` page is shown to the user, the value of the `access` attribute becomes temporarily true and the associated rules are evaluated.

Every rule defines a condition and several sets of actions. When the condition is fulfilled, the set of `trueActions` is applied. Otherwise, another set of actions (`falseActions`) is applied. In our case, there are two rules defined. When the condition `personal.learningStrategy == "TopDown"` is fulfilled, one action to be applied is defined. This action changes the value of attribute `suitability` of the `schema-1` concept (increasing its value by 50). The other rule changes the value of the `suitability` attribute of the `example-1` concept when the condition `personal.learningStrategy == "BottomUp"` is fulfilled. This means that a concept of programming scheme type is considered to be more suitable when the strategy ”from general to specific” is chosen and a concept of program example type is considered to be more suitable when the strategy ”from specific to general” is chosen. Since both rules specify the `isPropagating` attribute as true, by changing the value of the `suitability` attribute inside the `schema-1`/`example-1` concepts also the rules associated with these attributes are evaluated. Described mechanism observes the pages user is selecting and updates of the user model appropriately.

Presented approach of the user model definition in current adaptive hypermedia applications assumes the implicit semantics of attributes and their values. Only the designer knows what particular attribute stands for and what is the interpretation of its value. In order to use a user model by several adaptive hypermedia applications, an explicit definition of the semantics of the user model attributes and a separation of the domain model and the user model is required.

Separate user model gives rise to several problems. One of the main issues is related to the user model initialization. To achieve effective personalization, the adaptive hypermedia application needs to know user characteristics as accurate as possible. When a user approaches the system for the first time, the user model is usually ”empty” and for the majority of user characteristics default values can be only used. As the user works with the adaptive hypermedia application more and more, the user model becomes more accurate and personalization more effective.

The problem arises when the user employs more that one adaptive hypermedia application and these applications do not cooperate in the task of modeling user characteristics. An initialization phase is needed for every adaptive hypermedia application. This holds even in the case when the applications use domain independent characteristics such as presentation preferences, desired language, which do not change along with progress of the application usage. A limited solution would be to collect the knowledge about the user in a central knowledge base and use the knowledge by several adaptive hypermedia applications as appropriate. Adaptive hypermedia applications working on the same (or similar) domain would cooperate on the same part of the user model.

A need to understand semantics of data stored in a user model is crucial for effective collaboration between several adaptive hypermedia applications. Adaptive hypermedia applications need to know what the others collected about the user and effectively use it. Collaborating adaptive hypermedia application may discover new knowledge or at least refine the known one and thus affect the behavior of the other adaptive hypermedia application with the aim of accurate adaptation.

3. Shared User Model

Regarding to the problems mentioned in the previous section, we proposed an approach to support sharing the user model among several adaptive hypermedia applications. It is based on the following principles:

- the user model management is independent of the implementation of adaptive hypermedia applications, which employ the user model,
- the semantics of the data stored in the user model is explicitly defined,
- the user is able to observe/change information stored in the user model.

To achieve the independence of the implementation of an adaptive hypermedia application, we proposed to employ web services technology. The purpose of the *User Model Web Service* (UMoWS) is to store the user model and provide functionality to support cooperating adaptive hypermedia applications for obtaining information stored in the user model and for updating it. For the web services usage, the network environment is essential. Thus, the solution we proposed is generally suitable for cooperation between several web-based adaptive hypermedia applications.

To enable explicit semantics definition of data stored in the user model, we have proposed to use *ontology-based representation*. We use the semantic web technologies for ontology representation like RDF(S) or other higher level ontology languages like OWL. The existence of tools and query languages like RDQL gives us also a possibility to discover new knowledge from the stored characteristics, what may extend possibilities of the adaptation. Meta-model of the user model is presented in Fig. 1. Its basic building block is an RDF subject-predicate-object triple, which provides means for building the semantic model of the particular user.

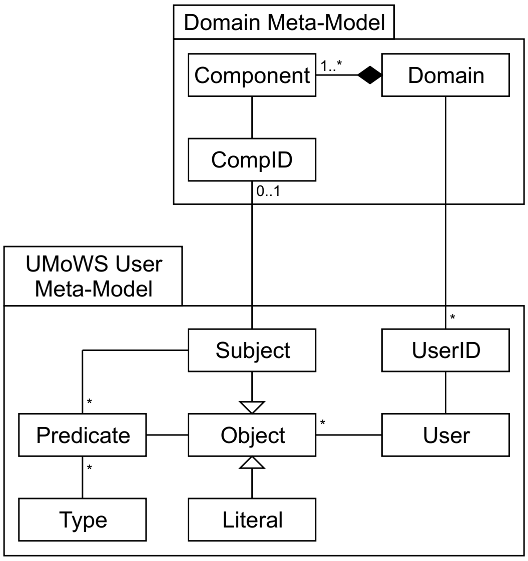

Figure 1. UML class diagram of ontology-based user model representation.

Ontology based representation of the user model described in the code fragment in Section 2 is presented in Fig. 2. Classes are depicted using solid lines and instances using gray lines. The relation between the class and its instance is expressed by the `io` (instanceOf) edge. The suitability attribute value bind with the concept assigned to a particular user is represented by the `Concept`-`Suitability`-`Person` relation. There is one instance of the `Concept` class for every concept and one instance of the `Suitability` class for every user including attribute's value. The instance of the `Suitability` class is related to the instance of the `Person` class representing particular user. Every user has assigned learning strategy (the `LearningStrategy` class).

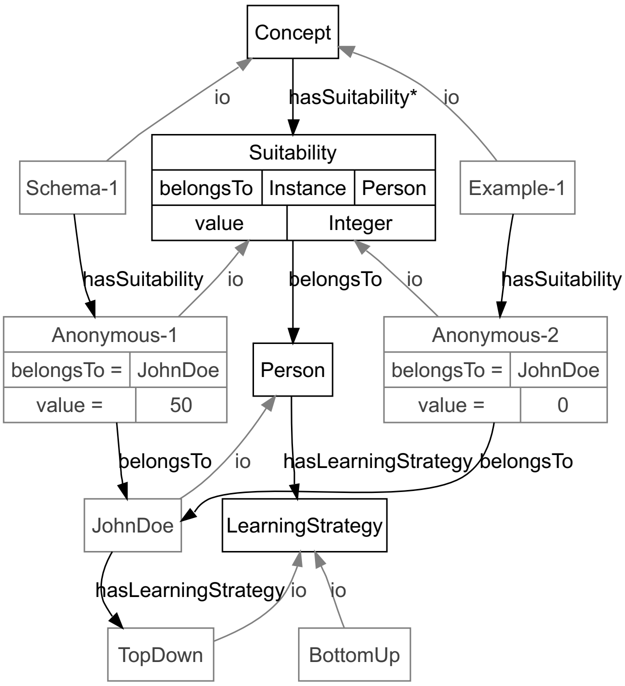

Figure 2. Ontology based representation of the user model.

Since adaptive hypermedia applications may differ in the domain they are operating over, there is a need to distinguish between the representation of the domain dependent part of the user model and the part which is domain independent and can be common for all (or several) domains. Domain dependent attributes usually express user characteristics, which affect the content or navigation adaptation (what information is presented and how can it be accessed), whereas domain independent attributes usually affect the presentation layout adaptation (how is the information presented). Fig. 1 depicts the domain dependency by relation between the `CompID` from the domain meta-model and the `Subject` from the UMOWS user meta-model.

The domain-independent aspects affecting the presentation adaptation can be the device a user operates for accessing required information, the language or media she prefers, etc. All these characteristics may affect the presentation no matter what domain is on side and all adaptive hypermedia applications may use/share them. For these characteristics common ontologies may be defined. One example of such ontology is Composite Capability/Preference Profiles (CC/PP) [8], a W3C recommendation, which can be used for expressing device capabilities and/or user preferences. Another example is an ontology for describing general user characteristics like contact information, emotional state, mental state, personality, etc. [6]

More demanding situation is when representing domain dependent user characteristics. Two different adaptive hypermedia applications operating over same (or similar) domain may use different levels of abstraction or other points of view. Thus, the same knowledge may be represented by different ways. The solution is to support the representations in different ontologies and if possible to provide the mapping between these ontologies by the UMOWS. Non-cooperating adaptive hypermedia applications can use the web service to store user characteristics in their own ontology and using only this ontology. Cooperating adaptive hypermedia applications would use also arranged ontologies.

Proposed architecture is shown in Fig. 3. Particular adaptive hypermedia applications communicate with the UMOWS by SOAP (Simple Object Access Protocol) protocol over HTTP (Hypertext Transfer Protocol) protocol using SSL (Secure Socket Layer). Every adaptive hypermedia application has its own domain/adaptation model and uses UMOWS to manage the user model. Users access adaptive hypermedia applications via web interfaces they provide (by HTTP protocol using SSL) using assigned user accounts. Since adaptive hypermedia applications need to know how to access web service, they provide an interface for configuring address of the web service server and credentials used for authentication/authorization when accessing web service.

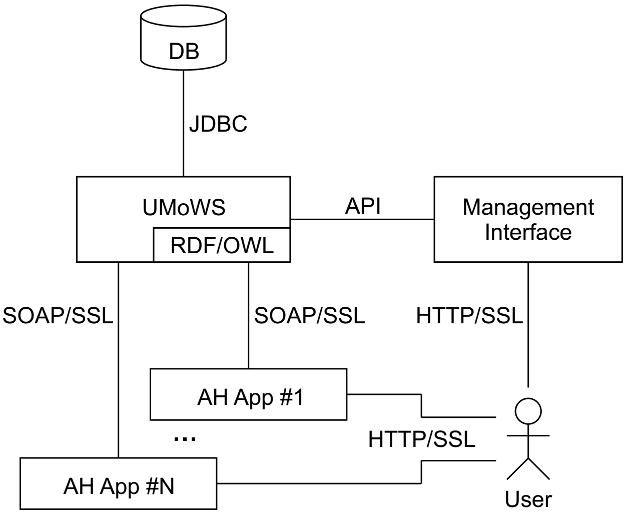

Figure 3. Architecture for sharing user models among several applications

To ensure a possibility for the user to observe/change the state of the user model, the web service provides appropriate interface for browsing the ontologies used for representing data in the user model and for changing the instances of these ontologies. We have included *Management interface* into the proposed software architecture for sharing user model among adaptive hypermedia applications for these purposes.

The user creates an account for the adaptive hypermedia system in the UMOWS through the Management Interface, which is realized as a web interface using UMOWS internal API. Management interface allows the user to define the access for cooperating adaptive hypermedia applications, the same as the privileges they have when accessing the web service. In case of using an RDF model, privileges definition includes the namespace in the RDF model and particular access rights (a set of rules, where rule determines whether accessing application is authorized/denied to read/write from/to specified namespace).

Since a part of the user model content has permanent character, a serialization mechanism is needed. We use relational database accessed via JDBC for storage permanent data.

4. Related Works

Personis server [7] presents a similar architecture for sharing a user model as proposed in this paper. It is also based on the web services technology, and provides cooperating adaptive applications with the functionality for retrieving stored data and for updating it. Stored data is based on *component-evidence-source* triplets, where component represents the attribute of the user model, evidence its value and source is the origin adaptive application which stored the information. The main focus is on the user control and scrutability of the user model. The main difference between this approach and our approach is in the representation of the stored data. In the Personis server no explicit definition of the semantics is considered. This fact limits the usage of stored information by other adaptive applications and overall cross-application adaptivity.

Another approach for sharing a user model is a *CompAS* architecture [2]. It alters the well known adaptive hypermedia reference models (e.g., AHAM and the Munich Model), by introducing the notion of services and semantic description of functionality in terms of ontologies. The key feature of CompAS is modularization of knowledge models and as a consequence allowing existing adaptive systems to share their user models by means of a centralized user model service. It exploits the agent based approach, where particular components act as agents using their own ontologies interconnected by the bridges. Partial verification of this architecture was done by extracting user model from the AHA! system and moved to ”u2m.org” UserModelService.

A distributed learner profile for the ELENA project [5] is based on the personalization services in a P2P learning network. Learning resources are available by resource providers forming resource provision network. Resource providers serve also the resource metadata, which are used by the course providers. Course providers could also provide course personalization service utilizing personalization metadata and learner metadata. Course providers can be used by the recommendation personalization services designated as the recommending and/or filtering services. Learners use the network by the personal learning assistents.

The main asset of our approach in comparison with mentioned approaches is the usage of generic standards like web services and semantic web technologies. The existence of the self-supporting UMOWS gives a possibility to map different ontologies used for representation of the user characteristics, so two different adaptive applications could work using two different ontologies and the UMOWS could ensure they are working using the same one.

5. Conclusions

In this paper, we have described our proposal of an architecture for sharing a user model among several adaptive hypermedia applications. Our approach is based on using separating user model from the domain model and explicit definition of the semantics of information stored in the user model. However, to make it use the shared user model, a significant modification of the current adaptive systems is required.

The main contribution of this approach is a possibility to share the knowledge about the user collected by one adaptive hypermedia application and use it by other adaptive hypermedia applications. This feature can help adaptive hypermedia applications to solve the problem with initialization of the user model. Besides that, central storage gives the user an opportunity to explore the stored information and to change it. Furthermore, user characteristics stored in the user model are represented using ontologies, what gives us a possibility to discover new knowledge.

References

[1] M. Bieliková and R. Habala. University course support by web-based adaptive e-Board. In F. Jakab, L. Samuelis, I. Sivy, and M. Bucko, editors, *Proceedings of International Conference ICETA 2004*, pages 395–402, 2004.

[2] V. Chepegin, L. Aroyo, P. de Bra, and D. Heckmann. User modeling for modular adaptive hypermedia. Technical report, SWEL'04 Workshop at the AH2004 Conference, TU/e Computing Science Report 04-19, 2004.

[3] P. de Bra, A. Aerts, B. Berden, B. de Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash. AHA! The adaptive hypermedia architecture. In *Proceedings of the ACM Hypertext Conference*, pages 81–84, Nottingham, UK, 2003.

[4] P. de Bra, G.-J. Houben, and H. Wu. AHAM: A Dexter-based reference model for adaptive hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots*, pages 147–156, Darmstadt, Germany, 1999.

[5] P. Dolog and W. Nejdl. Challenges and benefits of the semantic web for user modelling. In *Proceedings of AH2003 workshop at 12th World Wide Web Conference*, Budapest, Hungary, 2003.

[6] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. GUMO - the general user model ontology. In *Proceedings of International Conference on User Modeling UM 2005*, Edinburgh, UK, 2005.

[7] J. Kay, B. Kummerfeld, and P. Lauder. Personis: A server for user models. In P. de Bra, P. Brusilovsky, and R. Conejo, editors, *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 203–212, Málaga, Spain, 2002. Springer Verlag, LNCS 2347.

[8] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies*. W3C, 2001.

[9] N. Koch and M. Wirsing. The Munich reference model for adaptive hypermedia applications. In P. de Bra, P. Brusilovsky, and R. Conejo, editors, *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 213–222, Málaga, Spain, 2002. Springer Verlag, LNCS 2347.

[10] R. Kostelník and M. Bieliková. Web-based environment using adapted sequences of programming exercises. In M. Beneš, editor, *Proceedings of Int. Conf. on Information Systems Implementation and Modelling*, pages 33–40, 2003.