Abstract. In this paper we present an approach to sharing a user model between several adaptive hypermedia (AH) applications. In current AH applications, user model is often realized as the internal part of the system, without any possibility to share this model between other AH applications. Our approach introduce the web service acting as a central store of user characteristics represented in pre-defined ontologies defining semantics of the stored knowledge. Access and corresponding privileges to the web service is managed by user through the management interface. User is also able to inspect/modify the state of the user model.

1 Introduction

The aim of the adaptive hypermedia (AH) application is to 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. As described in reference models of adaptive hypermedia [1, 2], the user model together with the domain model and the adaptation model are three basic parts of every AH application. The role of the user model is to represent several users' characteristics such as user's knowledge, preferences, interests, tasks or goals. It is also considered as the source of the adaptation used in the adaptation model when adapting information from the domain model.

The user model is often managed by the AH application itself, realized as the internal part of the system, without any possibility to share this model between other AH applications. This approach has several disabilities.

* Supervisor: doc. Ing. Mária Bieliková, PhD., Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava
One of the main problems when working with the user model is the initialization of this model. To achieve effective personalization, AH application needs to know the characteristics of the user as much as possible. When the user is using the system for the first time, default values in the user model are used. As the user works with the system more and more, the user model is more accurate and personalization more effective. The problem is when the user uses more that one AH application and theses applications do not cooperate in the task of modeling user characteristics. Initialization phase is needed for every AH application, even they could use at least domain independent characteristics which do not (or not often) change such as presentation preferences, desired language, etc. Even better solution would be to collect the knowledge about the user in a central knowledge base and use this knowledge by different AH applications as appropriate. AH applications working on the same (or similar) domain would cooperate on the same part of the user model.

Next problem is that the user doesn't have the control or the information what is stored in the user model. AH application collects the information about the user regarding the interaction with system and uses this information later for the adaptive purposes. Collected information is often hidden to the user, what may cause in fear of misuse of this information. The skepticism may lead to refusing to use such systems. A solution to the problem of privacy would be to a transparent presentation of the contents of the user model with a possibility to change it. Only few existing AH applications support mechanism for inspecting/changing the state of 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 AH application is needed.

Regarding to the mentioned problems, an effort to separate the user model from the AH applications itself is the meaningful way towards the cross-application adaptation.

The paper is structured as follows. In section 2 we characterize the representation of the user model in current AH applications, together with the mechanism for updating this model and its utilization for the adaptation purposes. Next, in section 3, we describe the architecture for sharing a user model between several AH applications we proposed. Finally, we give conclusions and proposals for further work.

2 User Model in Current AH Applications

The source of the adaptation in AH application is the user characteristics significant from the adaptation point of view. It could be the user's knowledge of the domain AH application is working on, user's preferences, interests, goals etc. Generally, these characteristics are expressed by the user model.

From the modeling point of view, characteristics are usually represented as attribute-value pairs. There are 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 education area is an attribute representing the knowledge of a particular concept by the
user. Attributes independent on the domain represent characteristics useful without any consideration to the specific domain, e.g. desired language, preferred media, color scheme used in the presentation, etc.

User model is considered also as an extension of the domain model, where it extends the definition of the concepts with the definition of the domain dependent attributes. One example of such system is the system AHA! [3]. The definition of the user model lies in defining unique attributes for concepts. Values of the attributes can be booleans, integer numbers or strings. For every user, an instance of the user model is maintained and used for adaptation purposes.

From the maintenance point of view, following aspects should be covered when working with the user model:

- *initialization* of the user model,
- *user observation*,
- processing of the observed information in form of *updating* user model,
- *usage* of the user model for the purposes of the adaptation.

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

User observation consists of monitoring the user activity during the usage of the AH application. System monitors the links user is following, items she is selecting from the menu, time spent on particular pages, etc. Some AH applications prompt user to enter some kind of information. Educational AH applications sometimes provide the user with the possibility to enter whether she understands presented information.

The way observed information about the user is reflected in the user model, same as the way the state of the user model influence the adaptation, is defined in the adaptation model. Usually, it is realized by the rules. In system AHA!, the definition of concept contains the 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.

```xml
<concept>
  <name>de_koninck</name>
  <desc>De Koninck Beer</desc>
  <resource>de_koninck.xhtml</resource>
  <req>beer.interest > 50</req>

  <attribute name="access" type="bool" isSystem="true" isPersistent="false" isChangeable="false">
    <default>false</default>
  </attribute>
</concept>
```
In the previous code (adopted from [4], detailed explanation can be also found in [5]) a sample fragment of domain/user model definition in system AHA! is shown. Given code shows the definition of concept de_koninck representing information fragment about the Belgian beer brand. The content of the information fragment is defined in the page de_koninck.xhtml. The concept is considered to be suitable when the expression beer.interest > 50 is fulfilled (which means that user is interested in beers). Thus, the current value stored in the user model for the attribute interest defined in the concept beer is considered. Since the value differs for particular users, concept may be more or less suitable. This is the example of usage of the user model for the purposes of adaptation.

The definition of concept includes also the definition of attribute access which has associated rules (a set of generateListItem definitions) for updating user model. When the page de_koninck.xhtml is shown, the value of attribute access becomes temporarily true and the associated rules are evaluated. Every rule has defined condition. When the condition is fulfilled, a set of actions (trueActions) is applied. Otherwise, other set of actions (falseActions) is applied. In our case, when the condition beer.interest < 100 is fulfilled, one action to be applied is showed. This action changes the value of attribute interest inside the concept beer by increasing its value by 10. Since the rule has specified isPropagating attribute as true, by changing the value of attribute interest inside the concept beer, also the rules associated with this attribute are evaluated. Shown mechanism observes the pages user is selecting and updates the user model appropriately.

Presented approach of definition of user model in current AH applications assume the implicit semantics of attributes and its values. Only the designer knows what particular attribute stands for and what is the interpretation of its value. To use the user model by more AH applications, an explicit definition of the user model semantics and a separation of domain model and the user model is required.
3 Shared User Model

In the previous section, we characterized common used approach for modeling user characteristics in the under model. To support sharing of user model between several AH applications we proposed an approach based on the following principles:

- the user model management is independent on the implementation of AH applications which are using it,
- the semantics of the information stored in the user model is explicitly defined,
- user is able to observe/change information stored in the user model.

To achieve the independence on the implementation of AH applications, we proposed to use web services technology. The purpose of the User Model Web Service (WS) is to store the user model and provide functionality to cooperating AH applications for obtaining information stored in the user model and for updating it.

To ensure explicit semantics definition of the information stored in the user model, we proposed to use ontology based representation. Since AH 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 common for all 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 adaptation (how is the information presented).

The aspects affecting the presentation adaptation can be the device user is using for accessing the information, the language user prefers, suitable media, etc. All these characteristics may affect the presentation no matter what domain is on side and all AH applications may use/share them. For these characteristics a common ontologies may be defined. One example of such ontology is Composite Capability/Preference Profiles (CC/PP) [6], 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. [7].

More difficult situation is when representing domain dependent user characteristics. Two different AH 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 would be to support the representations in different ontologies and if possible to provide the mapping between these ontologies by the User Model WS. Not cooperating AH applications would use the web service to store user characteristics in their own ontology and using only this ontology. More intelligent AH applications would cooperate and use also arranged ontologies.

Our approach uses 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 the possibilities of the adaptation.
To ensure a possibility for user to observe/change the state of the user model, web service should provide appropriate interface for browsing the ontologies used for representing user model and for changing the instances of these ontologies. Our approach includes the Management interface used also for these purposes.

Proposed architecture is shown in Fig. 1. Particular AH applications communicate with the User Model WS by SOAP (Simple Object Access Protocol) protocol over http (Hypertext Transfer Protocol) protocol using SSL (Secure Socket Layer). Every AH application has its own domain/adaptation model and uses User Model WS to manage the user model. User accesses AH applications (by HTTP protocol using SSL) via web interfaces they provide using assigned user accounts. Since AH 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.

**Fig. 1.** Architecture for sharing a user model between several AH applications.

User creates an account for the AH system in the User Model WS through the Management Interface, realized as a web interface using User Model WS internal API. Management interface allows user to define the access for cooperating AH applications, same as the privileges they have when accessing web service. In case of RDF model, privileges definition includes the namespace in 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 some of the information stored in the user model has permanent character, a serialization mechanism is needed. Our approach uses relational database accessed via JDBC for storage permanent data.
4 Related Works

Similar architecture for sharing a user model is a Personis server [256]. 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 approach we proposed is in the representation of the stored data. In the personis server, no explicit definition of the semantics is considered, what limits the usage of stored information by other adaptive application and overall cross-application adaptivity.

5 Conclusions

In this paper, we have described architecture for sharing user a model between several adaptive hypermedia applications we approached. Our approval is based on using separating user model from the domain model and explicit definition of the semantics of information stored in the user model.

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

In the future, we plan to implement the prototype of the proposed architecture. We want to create the following subsystems:

• User Model Web Service managing user model storage, providing interface for AH applications to obtain the stored information and to update it,

• sample AH application using User Model WS for managing user model and

• Management Interface allowing user to define access rights for cooperating AH applications and an interface for observing/updating user model.

Acknowledgement: This work was supported by Science and Technology Assistance Agency under the contract No. APVT-20-007104.

References


