Broker-based Discovery Service for User Models in a Multi-application Context

Vadim Chepegin, Lora Aroyo, Paul De Bra
Eindhoven University of Technology, Computer Science Department
{v.i.chepegin, l.m.aroyo, p.m.debra}@TUE.nl

Abstract

In order to efficiently manage and use of the information on the Web it becomes crucial to provide personalization not only within single systems, but across various web applications. Interoperability of user models, on both syntactic and semantic level, emerges as a vital issue in order to achieve seamless personalization across different user-adaptive systems. In this paper we discuss the architecture of a Broker-based Discovery Service for User Models (BD-SUM), which allows applications to discover and invoke semantically described user models, in a multi-application context.

1. Introduction

In [1] the authors argue that for the effective work of a user in a multi-application environment, applications need to be aware of and able to react to the user's actions in external applications. Nowadays, in an educational context, a lot of effort is focused on the integration of intelligent tutoring systems with learning management systems [2], on the creation of educational web service networks [1] and on the use of Semantic Web technologies to achieve semantic interoperability and integration of various information sources [3].

The motivation for this is to achieve an extendable and scalable network of intelligent services, which can be discovered and invoked in order to satisfy users' goals, tasks or level of knowledge. In such a dynamic and versatile context, users need seamless adaptation across applications. To provide this level of adaptation, each application must be able to find, understand, and respond to the user's actions in other applications [1], [3].

2. User Modeling for Multi-application Context

Imagine the following situation: Jill visits the New National Gallery in Berlin and she is using her museum guide program on her personal digital assistant (PDA). While walking through the museum, her PDA guides her along the paintings she would be interested in and gives her additional information (usually coming from different sources) related to the paintings and painters she sees. The presentation of this information is personalized according to what the PDA knows about her so far – preferences, interests, knowledge and history of events. Typically, the PDA applications, including the museum guide, will store the information about Jill in separate user models. Additionally, Jill uses various web-based services in her work or leisure time, which also store information with respect to her preferences, interests in separate user models. Within the context of the CHIME project [3], we focus on the realization of examples like the one above. We propose a next generation of user adaptive systems (UAS), which will allow user model interoperability, in order to achieve the smooth communication and information sharing between various applications [3].

We consider the user model interoperability from two perspectives. Firstly, at a syntactic level, we refer to the ability of the systems to access and exchange user information when needed. Secondly, at a semantic level, we speak of the understanding and integrating of the user information from various user applications. Subsequently each application would be able to interpret and employ this information for its own purposes.

The *syntactic level* involves issues of information storage and retrieval. Researchers usually make ad hoc decisions, which serve the needs of their local systems quite well. The focus is primarily on the development of standards for communication protocols [2], and on information stores, like in Edutella project. Nowadays the syntactic level is served primarily by XML-based data exchange formats.

Within the CHIME project we follow the standardization efforts described in UserML/UserQL [4] and we define an orthogonal XML-based Markup Language for User Modeling (MLUM). It allows user adaptive systems to both represent, query and update their user models. In order to exemplify the use of this language we provide a web-service working with MLUM. It has been realized in Java and can communicate with other systems via SOAP. The *semantic level* makes explicit the meaning of content models and communication activities. In the current research results we distinguish three architectural solutions for semantic interoperability of user models: (1) Shared awareness. Each application within the framework shares the same view on the domain, the user's tasks, user modeling and adaptation [4], or a mapping is given manually (e.g. mapping from different activities to concepts they contribute in [2]); (2) Private awareness. Only the components that have created the models know how to interpret this information and how to reason with it [1]; and (3) Inferred awareness. The applications do not know a priori about the way others perform the user modeling, but they can automatically determine this with the help of explicit descriptions (ontological) of the semantics of the user model content and context.

In the context of the CHIME project we refer to a combination of the first and the third situations, exemplified by a Broker-based Discovery Service for User Models, which sets the ground for achieving both syntactic and semantic interoperability between user adaptive systems. In the next sections we will present some design and implementation aspects.

3. Broker-based Discovery Service for User Models

To be able to understand and exchange information with each other applications need to be aware of the context, purpose and methods which other applications use to create their own models. The Broker-based Discovery Service for User Models (BD-SUM) is an architecture for user modeling in a multi-application context, which allows various applications to discover and invoke semantically described user models. Our main goal is to enable user adaptive systems to work together and thus provide the user with a smooth adaptation across them. BD-SUM allows a strict separation and independence of the various knowledge and content models [3]. For this, BD-SUM defines the components: UM-Broker, reasoner, matchmaker, invoker, and pool of ontologies (e.g. domain ontologies, user modeling ontology and many other domain-independent). On the syntactic level BD-SUM builds upon UserML/UserQL [4] and uses an orthogonal Markup Language for User Modeling (MLUM) for the representation and querying of user models. On a semantic level BD-SUM uses ontologies to represent the meanings of the different user models. New applications register by the *UM-Broker* and provide it with the ontological description of their internal user model, domain and possible other application models.

This allows to search and find the most appropriate fragments of user models. The *Repository* is used to store the instances of all the application ontologies. The discovery of services is realized by the *Matchmaker* and *Reasoner* allowing for semantic matching between the service request description and the published service description. Queries with user model and application domain data (using the user modeling and domain ontologies) can be sent to the Repository. The invocation of discovered user models fragments involves sending to the requester messages which contain the fragment itself, and links to all available registered ontologies. We also work on a UserModelingOntology to support the semantic matching and search of user model fragments across all the registered applications. It is an upper level ontology, which does not specify concrete realizations of modeling users' characteristics.

4. Current and Future Work

Current efforts focus on the exemplifying the proposed syntactic level interoperability by allowing AHA! applications to exchange UM data with other web applications via the user modeling service. The formal description of the MLUM is also being finalized.

10. References

[1] Chr. Brooks, M. Winter, J. Greer, and G. McCalla. The massive user modelling system (mums). In *Proceedings of ITS 2004*, Springer LNSC 3220, pages 635–645, 2004.

[2] P. Brusilovsky. Knowledgetree: A distributed architecture for adaptive e-learning. In *Proceedings of WWW 2004 (Alternate track papers and posters)*, pages 104–113, 2004.

[3] Paul De Bra, Lora Aroyo, and Vadim Chepegin. The next big thing: Adaptive web-based systems. *Journal of Digital Information*, 5(1), 2004.

[4] D. Heckmann and A. Kr¨uger. A user modeling markup language (userml) for ubiquitous computing. In *Proceedings of UM 2003*, pages 393–397, 2003.