H AH 2008

5th International Conference on

Adaptive Hypermedia and Adaptive Web-Based Systems

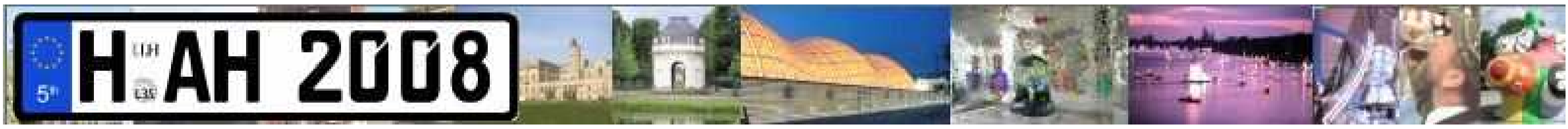
Hannover, Germany 29 July - 1 August 2008

User Model Integration

Workshop Proceedings

Organizers

Shlomo Berkovsky, *University of Melbourne, Australia*
Francesca Carmagnola, *University of Turin, Italy*
Dominikus Heckmann, *Saarland University, Germany*
Tsvi Kuflik, *Haifa University, Israel*
Antonio Krüger, *University of Münster, Germany*



Proceedings
of the 6th International Workshop on
Ubiquitous User Modeling
with the focus on:

User Model Integration
(UMI 2008)

in conjunction with the
5th International Conference on
Adaptive Hypermedia and
Adaptive Web-Based Systems

July 29th, 2008
Hannover, Germany

Shlomo Berkovsky, Francesca Carmagnola, Dominikus Heckmann,
Tsvi Kuflik and Antonio Krüger

Preface

Ubiquitous user modeling describes ongoing modeling and exploitation of user behaviour with a variety of systems that share their user models. These shared user models can either be used for mutual or for individual adaptation goals. Ubiquitous user modeling differs from generic user modeling by the three additional concepts: *ongoing modeling*, *ongoing sharing* and *ongoing exploitation*. Systems that share their user models will improve the coverage, the level of detail, and the reliability of the integrated user models and thus allow better functions of adaptation. Ubiquitous user modeling implies new challenges of scalability, scrutability and privacy.

A key challenge for ubiquitous user modelling is user models integration. Different systems may represent the same information in different ways, using different syntactic and conceptual structures. Therefore means for negotiation and clarification of data among systems are required. This is fundamental not only to access information, but also to reuse information.

Recently, suppliers of user profiles have shown an increased awareness of the need for standards for representing and exchanging user model data. At the same time we observe that dealing with syntactic and semantic heterogeneity of user models is pretty complicated, especially in an open environment like adaptive hypermedia and adaptive web-based systems. The issue is: how can semantic heterogeneity be handled for ubiquitous user modeling? How can the Semantic Web technologies be employed to cope with such heterogeneity?

These issues are the focus of the workshop. We deal with both theoretical and practical aspects of integrating user models from various sources, across different domains and different representations. The goal of this workshop is to bring together academic and industrial researchers from these communities to discuss the most innovative approaches to ubiquitous user modeling, to enhance the exchange of ideas and concepts, to determine the veins the research should proceed, and to go one step further towards personalization in ubiquitous computing. We expect that as a result of the workshop, new research directions will be defined and new collaborations among the workshop participants will be formed.

The papers accepted for presentation at the workshop represents the above diversity of topics in that area, they vary from abstract, theoretical frameworks to systems architectures, in various specific application areas.

The organizers: Shlomo Berkovsky, Francesca Carmagnola. Dominikus Heckmann, Tsvi Kuflik and Antonio Krüger

Acknowledgements

Many people contributed to the success of the workshop. We owe special thanks to the members of the program committee for their efforts in reviewing the workshop submissions.

Liliana Ardissono, University of Torino, Italy

Lora Aroyo, Free University Amsterdam, the Netherlands

Jörg Baus, DFKI GmbH, Germany

Federica Cena, University of Torino, Italy

Nadja De Carolis, University of Bari, Italy

Vania Dimitrova, University of Leeds, England

Peter Dolog, Aalborg University, Denmark

Cristina Gena, University of Torino, Italy

Judy Kay, University of Sidney, Australia

Alexander Kröner, DFKI GmbH, Germany

Andreas Lorenz, Fraunhofer Institut, Germany

Francesco Ricci, Free Univ. of Bozen-Bolzano, Italy

Ilaria Torre, University of Torino, Italy

Andreas Zimmermann (FIT), Germany

Table of Contents

User Model Integration in a Distributed Adaptive E-Learning System

Peter Brusilovsky[1], Sergey Sosnovsky[1], Michael Yudelson[1], Amruth Kumar[2], Sharon Hsiao[1]

[1]University of Pittsburgh, School of Information Sciences,
135, N. Bellefield ave., Pittsburgh, PA, 15260, USA
[2]Ramapo College of New Jersey
505, Ramapo Valley Road, Mahwah, NJ 07430-1680
{peterb, sas15, mvy3}@pitt.edu,
amruth@ramapo.edu, hyl12@pitt.edu
http://www.sis.pitt.edu/~paws/

Abstract. The integration of adaptive educational systems is changing from an interesting research problem into an important practical task. One of the major challenges that need to be accepted on the way is the development of mechanisms for student model integration. In this paper we propose an approach for aligning overlay models of students' knowledge collected by different educational tools relying on different domain representations. The approach is based on common protocols for evidence communication and manual mapping of underlying domain models. This approach has been applied for integration of two existing educational systems in the framework of undergraduate Java course.

1 Introduction

Over the last 10 years, a range of adaptive systems migrated from research labs to real life. Web recommender systems [1], mobile tourist guides [2], and adaptive e-learning systems [3] are now used by hundreds of real users. In some application areas the "density" of practical adaptive systems is reaching the point where several adaptive systems are available. Yet, in most of the cases, these systems do not compete, but rather complement each other offering unique functionality or content. This puts the problem of using several adaptive systems in parallel on the agenda of user modeling community. This problem has been explored over the last few years by several research teams, which considered it from several prospects: cross-system personalization [4], user model ontologies [5], and user modeling servers [6].

The main challenge of using several adaptive systems in parallel (or, as we may also say, a distributed adaptive system) is making the whole more then the sum of its parts. In this context, it means that each of the system should have a chance to increase the quality of user modeling and adaptation using integrated evidence about user, which was collected by all participating system. So far, it looks like the most popular approach to solving this user model integration problem is translation [7] or *mediation* [8] from one user model to another. However, in a number of cases translation of the whole user model from one representation to another can be a relatively costly approach. For example, if two adaptive systems are used not in a sequence, but in parallel, the user models on both sides are being constantly updated. Thus, to take the joint information about the user into account, each of the system will need to translate the user model from another system before every adaptive decision is made.

A very good example of this situation is a distributed adaptive E-Learning system. In such a system, a student can work with educational activities provided by several systems. Each of the involved systems receives evidences about user knowledge and attempts to build the student knowledge model. To make this model reliable, each of the involved systems should take into account evidences about the student produced during her work with all systems. However, as our experience of work with distributed system shows, a student can switch from one system to another many times even within a single session [9]. To avoid multiple translation from one user model to another within the same session, we explored an alternative approach to user modeling in distributed adaptive systems, which we call evidence integration. With this approach, adaptive systems do not exchange user models, but instead exchange evidences received while working with the same student. In this case, the problem of user model integration becomes the problem of evidence integration. While evidence integration is a relatively simple task in some domains (i.e., user's ratings for a specific movie can be easily taken into account by multiple recommender system), it is not the case in E-Learning. In E-Learning, each educational activity (i.e., problem, quiz, or example) is typically described in the terms of a system's internal *domain model*. Using this knowledge and information of student success or failure while working with the activity, the user modeling component updates student knowledge model. In a rare case where the component systems share the same domain model [10] integrating evidences from two or more adaptive systems is a relatively simple problem. However, in reality two adaptive systems developed for the same domain (such as Java programming of SQL) have very different domain model. In this case evidence integration become a real problem, which requires some kind of translation from one *domain model* to another.

This paper presents a practical example of distributed user modeling, which involves two e-learning systems with considerably different domain models for Java programming language. The goal of our work is to stress the problems of distributed user modeling in the field of E-Learning and provide an example of conceptual and architectural integration, which was used in a real college-level course.

The rest of this paper is structured as follows. Section 2 describes the details of two educational systems for Java programming domain developed by different research teams and relying on different domain models: Java-Problets and QuizJET. Section 3 discusses the problems we had to overcome while integrating these two systems on the level of their domain and user models. Section 4 presents the implementation details of the integration mechanism. Section 5 finishes the paper with discussion and plans of future work.

2 QuizJet vs Problets: Two Systems and Two Domain Models

2.1 Ramapo College's Problets

Problets (www.problets.org) are problem-solving tutors on introductory programming concepts in C/C++/Java/C#. They present programming problems, grade the student's answer, and provide corrective feedback. Problets sequence problems adaptively [11], and generate feedback messages that include the step-by-step explanation of the correct solution [12]. Students can use Problets for knowledge assessment and self-assessment, as well as for training problem-solving skills. The reified user interface of Problets emphasizes the use of mental models students need to maintain, when solving the problems [13]. Fig. 1 presents the student interface of a Problet on if/if-else Statements in Java. The bottom-left panel contains the simple Java program. The students need to evaluate the program and answer a question specified in the top-left panel. The system presents student's answers in the right-bottom panel, and indicates the correct and incorrect answers by marking them in green, and red correspondingly. The detailed help on how to use the system, submit the answers and read system's feedback messages can be always opened in the right-top panel of the Problet interface.

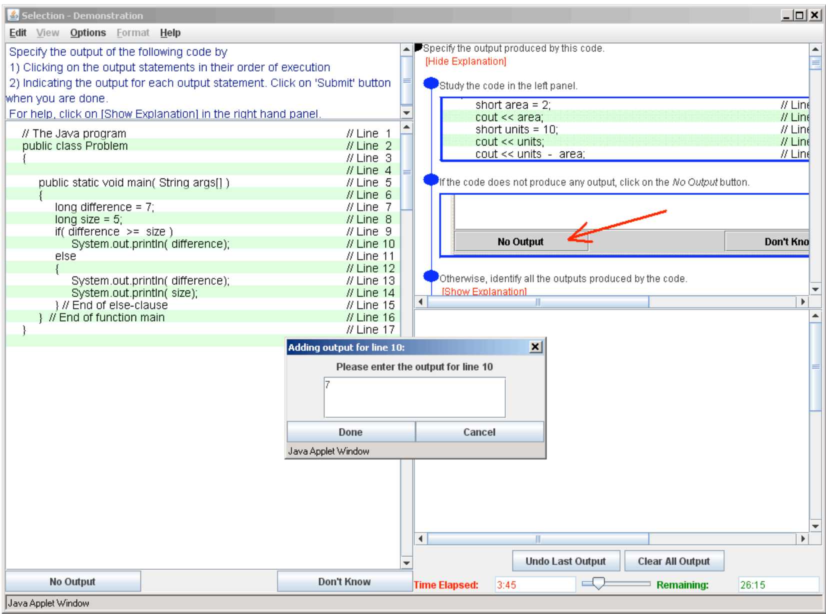


Fig. 1. A Problet on if/if-else Statements in Java.

Problets rely on the concept map of the domain, enhanced with pedagogical concepts called learning objectives, as the overlay student model [14]. Each learning objective is associated with the proficiency level calculated based on the student's answers. The student model provides the basis for adaptive decisions made by the tutor, by associating a proficiency model with each learning objective. The system propagates the proficiency values to the top levels of the concept hierarchy. At any moment of the tutoring session, a student can observe the current state of her/his user model. Fig. 2 demonstrates an example of the user model snapshot for the if/if-else Statements in Java.

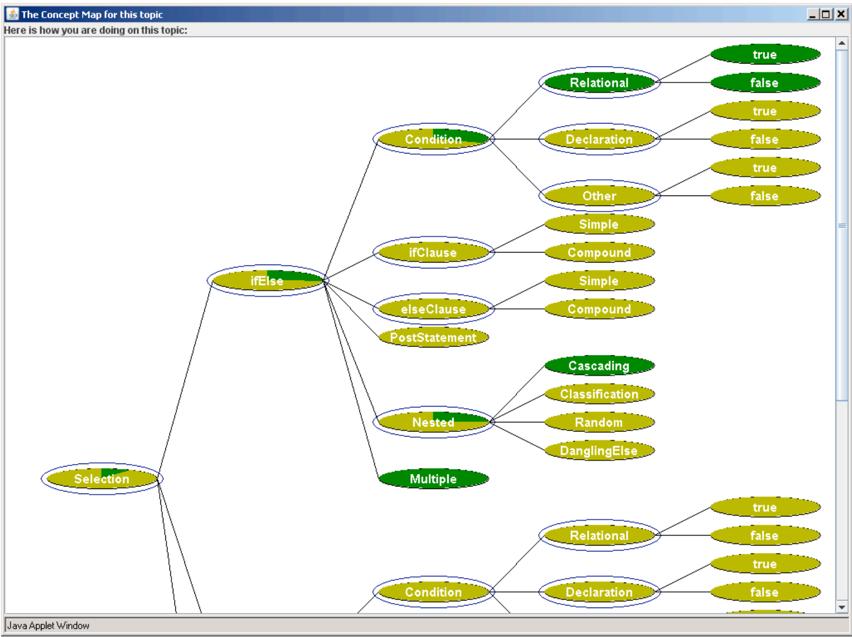


Fig. 2: A part of domain hierarchy on if/if-else Statements in Java. Learning objectives are associated with each concept in the hierarchy.

2.2 University of Pittsburgh's QuizJet

QuizJET (Java Evaluation Toolkit) is an online quiz system for Java programming language. It provides authoring and delivery of quiz questions and automatic evaluation of students' answers. A typical question in QuizJET is a simple Java program. The students need to evaluate the program code and answer a follow-up question, which can take one of two forms: "What will be final value of the marked variable?" or "What will be printed by the program to the console window?". Upon evaluation of the student's answer QuizJET provides short feedback specifying the correctness of the answer and the right answer in the case a student has make a mistake.

Fig. 3 demonstrates the student interface of QuizJET. The Java programs constituting QuizJET questions can contain one or several classes. For switching between classes QuizJet implements tab-based navigation. The driver class containing the main function (the entry point to the program) is always placed in the first tab, which also presents the question itself, processes the student's input and presents the system's feedback.

The important feature of QuizJet is parameterized questions. One or more numbers in the code of a driver class are dynamically replaced with a random value every time the question is delivered to a student. As a result, the students can practice QuizJET questions multiple times, and every time the question will be different and have a different correct answer.

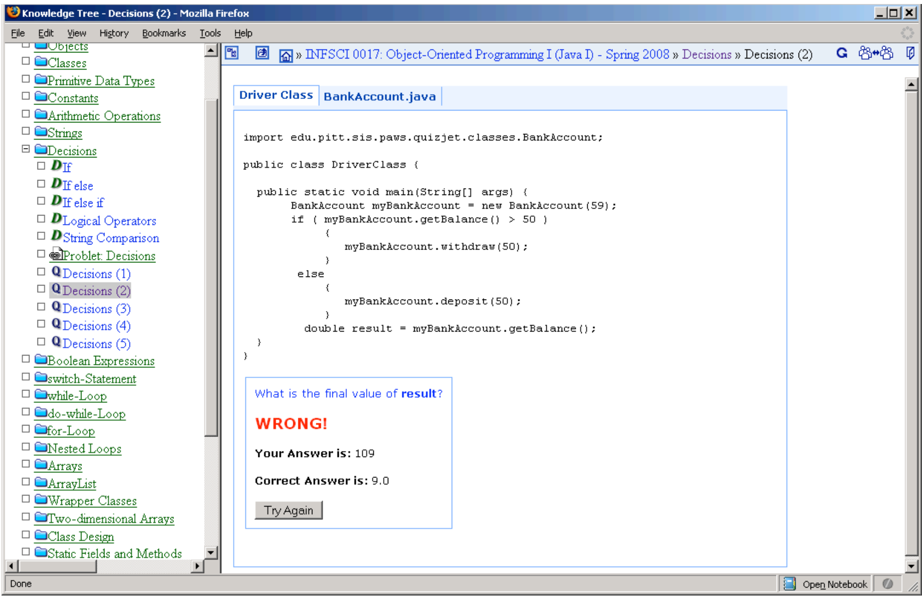


Fig. 3: An example of QuizJet question on Decisions in Java accessed through the Knowledge Tree Learning Portal.

Every QuizJET question is indexed by a number of concepts from the Java ontology. A concept in a question can play one of two roles: it acts either as a prerequisite for a question (if it is introduced earlier in the course), or as a question outcome (if the concept is first introduced by this question). Fig. 4 presents an extract from the Java ontology.

3 Integration: The Conceptual Side

Both Problets and QuizJet questions rely on conceptual content models that provide detailed representation of underlying domain knowledge. In order to provide consistent interpretation of the evidence reported by these two types of learning content, perform unifying user modeling and implement adaptive mechanisms taking into account student's work with both systems we need to integrate the underlying domain models on the level of concepts constituting them.

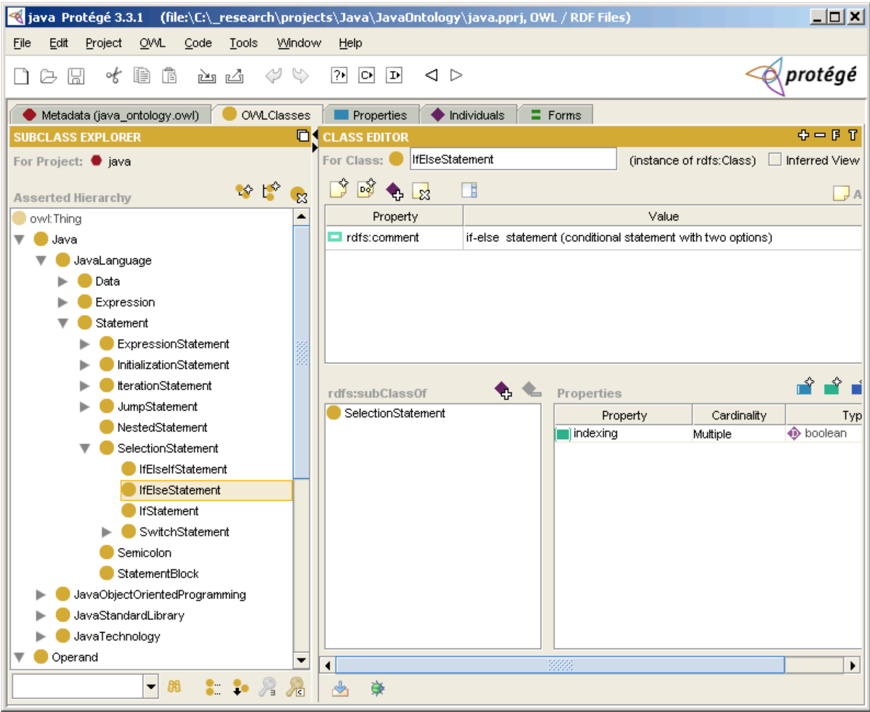


Fig. 4: An extract from the Java ontology.

The general task of domain model alignment potentially involves resolution of multiple model discrepancies on the two principle levels. The language–level mismatches, such as different syntax, expressiveness, or varying semantics of used primitives, need to be resolved in the first place. However, the more important are the model-level mismatches that occur due to the difference in structure and/or semantics of the domain models. Resolution of this kind of discrepancies involves dealing with such problems as:

- Naming conflicts (the same concept is defined in two models by different terms or the same term defines different concepts);
- Different graph structure (the models choose to connect relevant sets of concepts in different ways)
- Different scope (two models cover parts of the domain that only partially intersect or the scope of one model includes that of another model);
- Different granularity (the size of concepts differ across the models; a single concept of one model represents a piece of domain knowledge covered by several concepts of another model);
- Different focus (the models examine different modeling paradigms or adhere to different modeling conventions).

This list does not include the mismatches specific for the formal models employing advanced modeling primitives, such as typed relations and axioms (e.g. same entity can be modeled as a concept and as an attribute). In our study, we did not have to deal with this kind of discrepancies, neither we were resolving the language-level mismatches. Each of the problems listed above, however, occurred multiple times, while we were integrating the Java ontology and the learning objective models used in Problets.

Unlike QuizJet questions that are indexed with the concepts from the same ontology, each Problet relies on separate model of learning objectives. These models cover 6 large topics of Java programming language: (1)Arithmetic Expressions, (2) Relational Expressions, (3) Logical Expressions, (4) if/if-else Statements, (5) while Loops, (6) for Loops.

The combined scope of these topic models is several times narrower than the one of the Java ontology. At the same time the granularity of Problets' models is much higher. The total number of concepts in the Java ontology is about 500; the cumulative number of nodes in the Problets' models is more than 250. The most important problem we had to deal with is the difference on the modeling approaches (or different focus of modeling) used in Java ontology and Problets' domain models. Every learning objective models application of a concept in a particular learning situation (e.g. different objectives model the simple if clause in the if-else-statement and the simple if clause in the if-statement). In other words, a learning objective can be described as a concept put in a context. To properly map the context of a learning objective most of the time we had to connect one learning objective to several concepts from the Java ontology. To prevent too aggressive evidence propagation to the concepts modeling context of learning objectives, while mapping we also provided weights (from 0 to 1) that define how much knowledge of a particular concept define the proficiency of the learning objective. An example of mapping a learning objective into concepts is given by the Fig. 5. This terminal learning objective from the Selection topic defines the application of if-else statement, when the condition part of the statement evaluates to true value. To properly match this particular situation, we need to use three concepts from the Java Ontology. The assigned weights indicate that the main concept is still *IfElseStatement*, although the evidence of mastering this learning objective will slightly contribute to the knowledge of concepts *RelationalOperator* and *True*.

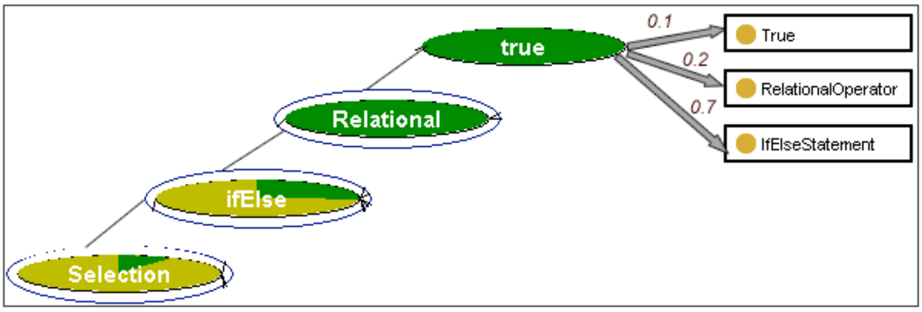


Fig. 5: An example of learning objective to concept mapping

4 Integration: The Implementation

To verify the practicality of the described approach we have administered a pilot study in the framework of an undergraduate Java course in the School of Information Sciences at the University of Pittsburgh. Both QuizJet quizzes and Ramapo College Problets are available to students via Knowledge Tree portal (Fig. 3). Knowledge Tree provides user authentication and authorization. It uses folder-document hierarchy to organize educational content. There are 6 Problets and 44 QuizJet questions are available to students of the introductory Java course taught at the School of Information Sciences, University of Pittsburgh.

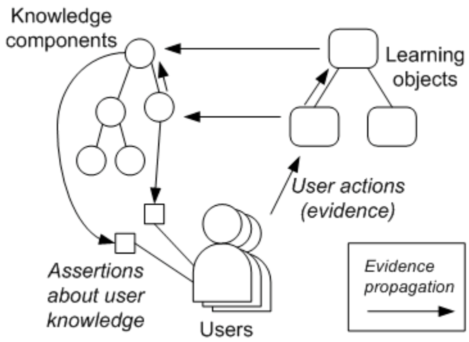


Fig. 6: Evidence propagation in CUMULATE [15].

Every time user submits an answer to the QuizJet question or Problet they notify user modeling server CUMULATE [15]. The notification is done via an HTTP request. Parameters of the request adhere to a simple protocol. Each report to user modeling sever should carry timestamp, group and user IDs, application and learning object the user worked with, and the result of the interaction (i.e., success, failure). Once evidence of new user action arrives, CUMLATE "propagates" it and updates overlay domain model of user knowledge. Special inference agents perform such propagation. Each agent is configured to deal with evidence from one or more external applications. Once new evidence arrives an agent is activated. For each learning object the agent retrieves corresponding metadata – concepts of the domain model (aka knowledge components). Based on the weights of the concepts for a particular learning object result of the interaction the agent updates assertions regarding user mastery of the concepts (Fig. 5). Note that the evidence propagation approach to user modeling applied by CUMULATE allows to take into account user work with both adaptive and non-adaptive learning objects – as long as concept-level metadata for these objects is provided by the content developer or any other stakeholder.

In the case of QuizJet and Ramapo Problets, the learning activities served by the systems are indexed with concepts of Java domain ontology described in section 3. This index along with weights (significance of each concept for the learning object) is cached by CUMULATE. The same inference agent handles evidence coming from both QuizJet and Ramapo Problets. This agent increases the mastery level of the corresponding concepts in the case of successful interaction (correct answer) and does nothing otherwise. Since ontology mapping ensures that QuizJet and Problets use the same pool of domain concepts, successful interaction with quizzes and Problets have an identical effect on the level of student knowledge of domain concepts. Users work with either QuizJet quizzes, Ramapo Problets, or a combination of the two is reflected in the user model. The current "integrated" state of the user model is available for all approved components (including QuizGuide and Problets) and can be used to provide a more precise adaptation.

5 Discussion

In this paper we have proposed a lightweight solution for integration of user modeling information collected by different educational systems. The resulting infrastructure allows two applications developed by different research teams and relying on considerably different domain models to be used by the students of the same course. The applications separately collect the evidence about student knowledge and communicate it to the user modeling server, which allows us to support more holistic user models.

As the first step of the further investigation of this approach, we plan to evaluate the results of the pilot study conducted in the undergraduate Java course. The focus will be made on the assessment of the predictive validity of the obtained user models.

Although, the reported evidence is not currently used for adaptation, the necessary protocols have been implemented on the user modeling side. The Ramapo team is modifying the Problets in order to retrieve the aggregate user models from the CUMULATE server. On the University of Pittsburgh side a similar role will be played by the QuizGuide adaptive service [16].

We are also going to apply this solution in other learning domains (C, C++, SQL) and integrate with other adaptive systems, e.g. Canterbury University's constrained-based tutors [17] and University of Malaga's SIETTE [18].

References

1. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) 291-324
2. Krüger, A., Baus, J., Heckmann, D., Kruppa, M., Wasinger, R.: Adaptive mobile guides. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) 521-549.
3. Brusilovsky, P., Peylo, C.: Adaptive and intelligent Web-based educational systems. International Journal of Artificial Intelligence in Education 13, 2-4 (2003) 159-172
4. Denaux, R., Dimitrova, V., & Aroyo, L. Integrating Open User Modeling and Learning Content Management for the Semantic Web. *In: L. Ardissono, P. Brna & A. Mitrovic*

(eds.), Proceedings of the 10th International Conference on User Modeling (UM'2005) (pp. 9-18), Edinburgh, Scotland, UK: Springer.

5. Dolog, P., & Nejdl, W. (2003). Challenges and Benefits of the Semantic Web for User Modelling. *In: P. De Bra (ed.), Proceedings of the AH2003 workshop*, Budapest , Hungary.
6. Niederée, C., Stewart, A., Mehta, B., & Hemmje, M. (2004). A Multi-Dimensional, Unified User Model for Cross-System Personalization. *In: Proceedings of the Workshop on Environments for Personalized Information Access at AVI'2004 (pp. 34-54)*, Gallipoli, Italy, from http://www.di.uniba.it/avi2004/e4pia/EPIA2004_proceedings.pdf
7. Sosnovsky, S., Dolog, P., Henze, N., Brusilovsky, P., Nejdl, W.: Translation of overlay models of student knowledge for relative domains based on domain ontology mapping. In: Luckin, R., Koedinger, K.R., Greer, J. (eds.) Proc. of 13th International Conference on Artificial Intelligent in Education, AI-ED 2007. IOS (2007) 289-296
8. Berkovsky, S., Kuflik, T., Ricci, F.: Cross-technique mediation of user models. In: Wade, V., Ashman, H., Smyth, B. (eds.) Proc. of 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2006). Lecture Notes in Computer Science, Vol. 4018. Springer Verlag (2006) 21-30.
9. Brusilovsky, P., Sosnovsky S., Lee, D.H., Yudelson, M., Zadorozhny V., & Zhou X. (2008). An open integrated exploratorium for database courses. In: E. Menasalvas & A. Young (eds.) *Proceedings of 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'2008), Madrid, Spain, June 30 – July 2, 2008* (accepted).
10. Trella, M., Carmona, C., Conejo, R.: MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educational Systems for the Web. In: Proc. of Workshop on Adaptive Systems for Web-based Education at 12th International Conference on Artificial Intelligence in Education, AIED'2005. IOS Press (2005) 27-34.
11. Kumar, A.N. A Scalable Solution for Adaptive Problem Sequencing and its Evaluation. In Proceedings of The 2006 International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2006), Dublin, Ireland, June 21-23, 2006.
12. Kumar, A.N., Explanation of step-by-step execution as feedback for problems on program analysis, and its generation in model-based problem-solving tutors, Technology, Instruction, Cognition and Learning (TICL) Journal, Vol 4(1).
13. Kumar, A.N., A Reified Interface for a Tutor on Program Debugging, Proceedings of Third IEEE International Conference on Advanced Learning Technologies (ICALT 2003), Athens, Greece, 7/9-11/2003, 190-194.
14. Kumar, A.N. Using Enhanced Concept Map for Student Modeling in a Model-Based Programming Tutor. International FLAIRS conference on Artificial Intelligence, Melbourne Beach, FL, May 11-13, 2006, 527-532.
15. Brusilovsky, P., Sosnovsky, S. A., & Shcherbinina, O. (2005). User Modeling in a Distributed E-Learning Architecture. Paper presented at the 10th International Conference on User Modeling (UM 2005), Edinburgh, Scotland, UK, July 24-29, 2005
16. Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2004). QuizGuide: increasing the educational value of individualized self-assessment quizzes with adaptive navigation support. In J. Nall, & R. Robson (eds.) *Proceedings of World Conference on E-Learning (E-Learn 2004), Washington, DC, USA, November 1-5, 2004*, 1806-1813
17. Mitrovic, A., Martin, B., Suraweera, P. Constraint-based tutors: past, present and future. IEEE Intelligent Systems, special issue on Intelligent Educational Systems, vol. 22, no. 4, pp. 38-45, July/August 2007.
18. Conejo, R., Guzman, E., and Millán, E. (2004) SIETTE: A Web-based tool for adaptive teaching. International Journal of Artificial Intelligence in Education 14 (1), 29-61.

User Model Interoperability in a SOA Environment

Federica Cena and Furnari Roberto

Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, Torino, Italy
{cena,furnari}@di.unito.it

Abstract. The paper presents an approach to achieve User Model (UM) interoperability among UM-based applications on the Web. In order to enhance the interaction capabilities of such systems, we use Web Service technologies for syntactic interoperability and Semantic Web languages for semantic interoperability, together with negotiation techniques based on dialogue. The paper describes a Service Oriented Architecture(SOA)-based framework where a central UDDI registry, enhanced with UM specific capabilities, is used to support and promote the cooperation between UM-based applications.

1 Introduction

Interoperability of User Model (UM) knowledge plays a crucial role when services access take places in an open and dynamic environment like the Web. The opportunities both for the final users and for the applications for sharing user knowledge are well known: on the one hand, users do not need to waste time training each new system they deal with, on the other hand, the adaptive systems are able to reach a deeper understanding of users. In fact, sharing User Models enables applications to cover more aspects of the user profile, increasing, at the same time, the level of detail and the reliability of user data. All this additional information about the user leads to adaptation results which are more appropriate for her [14].

In realizing interoperability, a primary problem is how a system can find other systems available to cooperate (*discovery* issue). Once a partner has been found, interoperability implies the ability to i) exchange information, which requires to agree on the format of the exchanged data (*syntactic interoperability*) and ii) to use such knowledge, which requires to correctly interpret the data (*semantic interoperability*). The UM exchange, in particular, could require non trivial pattern of communication between the requestor and the provider. For instance, if the systems do not share the same knowledge models, they may need to clarify the requested user feature (e.g. the provider is not able to correctly understand the requested information), or they may need to negotiate about the response when the exact one is not available or is not considered satisfying by the requestor. In these situations, the interacting systems could use some form of *conversation*, a complex interactions among two parties, which may evolve in different ways, depending on the state and the needs of the two participants.

The paper proposes a solution to support the UM interoperability process on the Web, exploiting Web standards for interoperability, i.e. Web Service technologies for syntactic interoperability, and Semantic Web paradigm for semantic interoperability. On

the one hand, the loosely coupled structure and the well accepted standards underlying Web services represent a solution to the integration of UM knowledge and personalization services from various systems, for the sake of cross-system personalization. On the other hand, resources from the Semantics Web, such as ontologies, metadata and specification languages, can provide semantic interoperability, since they are based on open standards and have been developed to allow common understanding among distributed systems in open environments.
In our work we propose a framework, based on the main principles of Service Oriented Architecture (SOA), which can be exploited by applications that want to share UM knowledge. We enhanced the SOA paradigm to handle the peculiar needs of the UM interoperability context, giving a sound environment where it is possible to implement existing approaches for interoperability (such as the dialogue approach of [5]) and new solutions.

The paper is structured as follow. We first present in Section 2 a scenario motivating the relevance of the discussed problem. Then, in Section 3, we present our approach, describing in details the proposed framework. Section 4 presents the framework in action dealing with the scenario. Section 5 concludes the paper discussing the benefits of our solution in comparison to existing solutions.

2 Motivating scenario

UbiquiTO [6], a mobile tourist recommender system for the city of Turin, lacks some important information about its user Mary. Thus, it is not able to provide an effective personalization. A possible solution (avoiding to bore Mary by directly asking her for the missing information) is to exchange information about her with other UM-based systems working in a similar context. First of all, UbiquiTO should find out systems which store such information. With the existing solutions, UbiquiTO has no means to automatically discover other systems. This means that UbiquiTO's developers need to explicitly encode this information in the system. For example, UbiquiTO knows about iCITY [4], a social recommender of events occurring in Turin. In order to start a collaboration, it has to contact it, and to reach an agreement on the protocols to be used in the communication. Since UbiquiTO and iCITY do not share exactly the same representation of the domain, some interpretation problems may occur. For example, when the requested user feature is not represented in a common way in the two systems, some form of conversation can be useful to clarify the meaning of the used terms.

As it can be seen in the scenario, one of the main obstacles for interoperability is that an application needs to discover other applications able to interoperate (i.e. the *discovery issue*) and needs some mechanisms and protocols that must be used to interoperate (i.e. the *syntactic interoperability issue*). Theoretically, all the agreements have to be taken in a Peer-to-Peer way, which is time consuming and requires a big overwork when dealing with a large number of services.
Another problem is related to the *semantic interoperability issue*: if the knowledge model is shared, and thus the involved services agree on the meaning of terms, it is possible to exploit simple *atomic communication* (where a system merely asks for the value of a property and the other system provides it) by means of standard *request/response*

invocation. Instead, when the knowledge models are note shared, atomic communication is not sufficient since systems have to agree on the meaning of each single property. *Conversation* can be used in order to negotiate the meaning (i.e. finding similar concepts, reasoning on concepts in order to find an agreement [10]) and also to approximate the response when an exact one is not available. The requirements for atomic communication and conversation are different: for the former, systems must only know how to ask for the desired property; for the latter, in addition to the previous information, systems must also know how to structure the conversation, how to express a message, and the allowed order of messages.

3 The SOA Framework

Taking into account previous considerations, in our vision a framework supporting UM interoperability has to offer tools and facilities to: i) support the discovery of systems offering the desired information; ii) provide shared ways to describe the different interoperability capabilities of applications (both syntactically and semantically); iii) support *atomic communication* and *conversation*. Moreover, it is important to respect the privacy policies of the involved services, protecting the UM features and the sensible data.
The framework we propose has the following features. It is based on Web Services and Semantic Web standards and it provides a centralized and shared definition of the tools supporting communication. It uses a UDDI (Universal Description Discovery and Integration) [1] registry which stores information about all the available services. In this central registry each service declares, beside its WSDL (Web Service Definition Language) interface, the supported syntactic and semantic communication tools. To take part in the framework, applications must use Web Service standards as a basic communication protocol. They must expose their services by means of a public interface described in WSDL, accessible via HTTP by means of SOAP (Simple Object Access Protocol) messages[1]. Systems have also to refer to an external ontology regardless of the inner knowledge representation. Finally, systems are required to use a mechanism like OpenId (http://openid.net) to provide a common user identification.

3.1 Conversation model

A novelty of the framework is the ability to support complex interaction in form of conversation. The framework is designed to support the implementation of complex interactions using different techniques and models. In this paper we exploit the dialogue model presented in [5] (based on Dialogue Games [11] and Speech Acts theory [12]), since it was specifically conceived for UM interoperability. This dialogue model adapts the diagnostic learning dialog model of [8] to UM interoperability context. The basic dialogue primitive in the model is the Speech Act (SA). Each SA is represented as a couple <*move, statement*>. A *move* is a domain-dependent verb expressing the system intention (e.g. to inquire, to deny, to accept, to inform, etc). A

[1] Simple Web Service wrapper can be used to provide WSDL interface to existing UM-based applications.

statement about the User Model is represented as a triple <*property,value,belief*>: for example, <*interestArt,0.8,0.2*> means that the value of the property "interest in art" is high with low certainty.
The building blocks of the model are the *dialogue games*: templates describing the communication behaviour the systems can follow to reach a particular goal. A dialogue game is activated when its *preconditions* are satisfied and is defined by the combination of:

– *conversation protocol*: conversation expressed in terms of messages exchanged between two roles, i.e. the allowed moves in SA and how to order them, from the different point of view of the requestor and responder;
– *focus strategies*: strategies to collect the concepts that can be discussed in a conversation;
– *scope strategies*: strategies to select from the focus the concepts to be used as statements in the SAs;

The model was instantiated in the UM interoperability context in [5], where three main goals were identified: i) to clarify the request; ii) to approximate the response; iii) to explain the response. Three dialogue games were then defined to reach such goals.
The *Clarification Game* supports the goal of refinement of the request. The responder can use this game to restrict the request's context to disambiguate the concept, when systems do not share the same knowledge model. The rationale of the game is the use of concept properties to disambiguate the meaning of the requested User Model feature.
The *Explorative Game* supports the goal of approximation of concepts by collecting information about the concepts and relations in the knowledge base to find an approximate answer, when an exact one is not available or it is but the associated belief is too low. The rationale of the game is that if there is not the value of the requested concept, the values of related concepts can be used instead.
The *Explicative Game* clarifies why a particular user feature value or belief is present. This dialogue game can start when there is a discrepancy in the participants' beliefs that need justification.

3.2 Overview of the framework architecture

Each UM-based application running in our framework is a *Web Service* (see figure 1) and has to provide the basic WSDL operation to support atomic communication to share UM knowledge (e.g. *getValueOf(property,..)*, and the interface to correctly interact with a central registry, *Enhanced User Model UDDI Registry* (*EUMUR*). This is a an UDDI registry (used as a standard discovery tool) adapted and enhanced according to the peculiarity of the UM context. Beside the declaration of all the services cooperating in the framework, here we can find the definition of all the tools that can be used as model for the communication between services. EUMUR has three main components: the *Communication Tools*, the *Services Declaration*, and the *Search Network Buffer*.

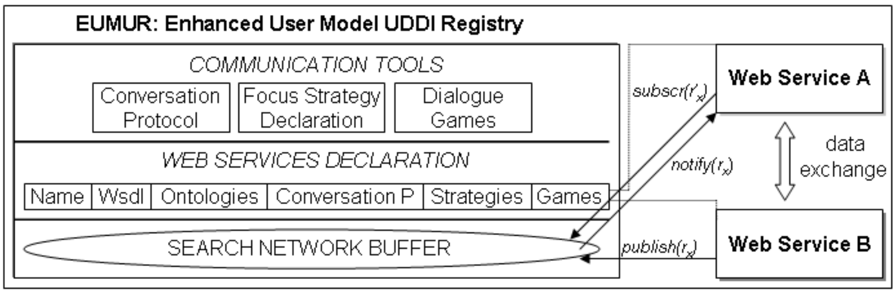


Fig. 1. Framework Architecture

Communication Tools. This component contains the list of supported mechanisms for enable different kinds of enriched interactions.
Conversation Protocol stores the description of all the conversation protocols used in the framework. For each conversation we have: (i) *convID*, a unique identifier in the framework; (ii) *role1 Interaction Protocol*, describes the complex sequence of messages that a Web Service playing the role1 has to implement in order to respect this conversation protocol; (iii) *role2 Interaction Protocol*, describes the sequence of messages from the role2 perspective.
Focus Strategy Declaration describes strategies to browse through ontology concepts. In particular we have: (i) *focuStratID*, a unique identifier; (ii) *strategy*, described as semantics-based query over the (RDFS or OWL) ontology.
Dialogue Game, the schema for a dialogue game (see section 3.1) that contains: (i) *gameId*, a unique identifier; (ii) *precondition*, a set of conditions that make the implementation of this game significant; (iii) *convId*, a reference to the Conversation Protocol used in this game; (iv) *focusStrategyId*, a reference to the Focus Strategies exploited in this game; (iv) *scopeStrategies*, the list of strategies used in this game to select between the concepts retrieved by the focus strategies.

Web Services Declaration. It contains the list of Web Services available in the framework associated with the declaration of the supported *Communication tools*. For each UM-based Web Service we have the following fields: *ServiceName* (the name of the service); *Wsdl reference* (the reference to the WSDL file describing the operations offered by the service); *Ontologies* (the list of all the ontology schemes the service refers to); *Conversation protocol* (the list of conversation protocols the service can implement); *Focus strategies* (the list of focus strategies the service is able to exploit); *Games* (the list of dialogue games the service is able to play).

By means of this central registry an application looking for partners to exchange user knowledge can immediately know if each candidate Web Service shares the same ontology, and which kind of protocols, strategies and games it supports. Note that the *Conversation Protocols* and the *Focus Strategies* are conceived as independent tools, that can be implemented also out of the context of a Dialogue Game.

Search Network Buffer. It is a shared network space able to automatically match service requestors with service providers of specific UM knowledge. The interaction model is managed according to the *publish/subscribe* pattern and the data exchanged represent requests of user features and responses of availability of values for the requested features. In details,

- all the services *subscribe* to the SNB asking to be notified (as a provider) when a certain kind of request arrives into the buffer (*subscr*(r'_x) in figure 1);
- when a service (acting as a requestor) looks for some user information, it *publishes* a request to the buffer describing the desired feature (*publish*(r_x) in figure 1);
- the SNB notifies (*notify*(r_x) in figure 1) all the subscribed services according to the request features;
- all the notified services look at the received request, and, if able to satisfy it, they reply to the SNB declaring their availability (just declaring that they have the requested data);
- the requestor service *reads* from SNB which are the available services, checks the *Service Declaration* to know the features of those services, and it can directly contacts each provider service to ask for the desired information.

In the SNB neither UM *dimensions* nor *values* are shared, since the buffer just hosts requests (and answers) of collaboration. The exchange of UM dimensions and values will take place in a peer-to-peer way, so, the requestor service is free to select which tools to use for the interaction (according to its internal policies), while the provider can apply its own privacy policies for data access.
The format of a SNB request is: *(Sender, Action, Kind, Ontology, Focus, Object, User)*, where *Sender* is the name of the requestor, *Action* is the constant *inquiry*, *Kind* is the typology of the request: *byUri* (if it refers to a specific ontology) or *byLabel* (if expressed by means of a label); *Ontology* keeps a reference to the supported ontology; *Object* contains the *URI* or the *Label* of the requested concept; *Focus* contains the focus strategy id used by the requestor uses; *User* refers to the user the request is referring to. For instance, when UbiquiTO needs information about the concept *conc#134* referring to the ontology *TourismTO*, it invokes the following operations on the SNB: *publish(iCITY,inquiry,byUri,TourismTO,null,conc#134,user456)* and in this case all the services previously subscribed by means of operations like *subscribe(MyName, inquiry, byUri, TourismTO)* will be notified by the SNB.

4 The Framework in Action

In this section, we applied our model to the scenario depicted in section 2. Each service in the scenario should respect the framework requirements, i.e. to provide a WSDL interface, to declares the ontologies, the interaction protocols, the focus strategies and the dialogue games it supports in the *Service Declaration* in EUMUR. Furthermore, each service must be subscribed to the SNB to publish and read information from it. UbiquiTO, to provide personalized services to its user Mary (*us897*), needs her interest[2] in the concept *conc#345B* of the domain ontology *TourismTo*. In order to ver-

[2] We assume that all the applications involved in the framework refer to the Public General User Model Ontology (GUMO, http://www.ubisworld.org/) for the definition of the UM features.

ify which systems may provide it, UbiquiTO publishes a request into the SNB: *publish(UbiquiTO,inquiry,byUri,TourismTo,null,conc#345B,us897)*. iCITY, subscribed for this kind of request is notified about the request and, since it has the requested value, answers to the call. UbiquiTO, looking at the registry, retrieves the iCITY WSDL reference, so it can directly ask it for the needed user feature, by means of the operation *getValue(byUri,conc#345B)*.

In a second moment, UbiquiTO, since it has to suggest to the user some relevant artistic places to visit, needs to know the user preference about the concept labeled as *church*[3] in its ontology *Art* . Since no systems share such ontology, it submits to the SNB the *byLabel* search request: *publish(UbiquiTO,inquiry,byLabel,art,null,church,us897)*. iCITY, subscribed to this kind of requests, answers offering its availability since it has the label *church* in its ontology *Tour-guide*; thus iCITY contacts UbiquiTO. However in Tour-guide the label is associated to two different concepts: *church* as place to visit and *church* as place for religious celebration. Looking at the Service Declaration, iCITY finds that UbiquiTO is able to play the *Clarification Game* (see 3.1) (clariG in the Fig.2) that can be used to refine the request. Thus it asks UbiquiTO to start an instance of this game: iCITY, following the focus strategy prescribed by the game, inquires for the distinctive properties of the concept labeled as *church*. For instance, iCITY could produce the SA <*UbiquiTO, UbiquiTO, inquiry, celebration_time*> since *celebration_time* is a discriminating feature for religious places. If UbiquiTO replies with a <*UbiquiTO, iCITY, deny*>, iCITY may ask for other relevant properties (such as form of worship, priest's name, etc), in order to be able to deduce which concept the label refers to.

5 Conclusion and Related Work

The paper deals with the interoperability among UM-based applications in order to exchange user model knowledge. The main contribution of our work is the proposal of i) an environment based on Web Services and Semantic Web standards, ii) a powerful discovery mechanism, ii) the support of conversations as means of negotiation. Two main kinds of UM interoperability solutions have been proposed in the literature: centralized [15] and decentralized [14]. Our intention is to exploit the advantages of the distributed approaches (such as flexibility in managing privacy) [14] providing a central shared point used as a warranted reference to cooperation. A similar model to the Search Network Buffer has been proposed in [7] in order to share UM fragments by means of a central repository. Other approaches propose a similar solution in a totally decentralized perspective, such as [13], applying agent-based technologies in ubiquitous environment.

We propose a mixed solution where the publish/subscribe pattern is just used as a central point for automatic user feature discovery, and where there is not a shared user model description and the exchange of user knowledge takes place in a peer-to-peer way. Another interesting mixed solution for UM interoperability is proposed by [3]. The author suggests the exploitation of a central registry to find out those web systems storing the requested user feature. Systems are required to make each UM available to

[3] In this example we suppose that applications use a shared vocabulary of terms.

other systems through a RDFS representation. The exchange of UM *values* is held according to a p2p modality. To solve semantic heterogeneity, systems lead a Similarity Measure Algorithm, which follows the word sense disambiguation theory to compute the semantic similarity among user model data.
Ontologies as basis for interoperability are used in particular in the learning environment. We can mention [2], which proposes an ontology-based framework where the UM exchange among several UM Servers is realized by means of a set of Ontology Server. The usage of ontology is also exploited in [9], which presents an architecture where a central general repository (GUC) maintains user models described by means of different user application-views, i.e.UM instances associated with several schemes. The issue of semantic interoperability is reached by means of facility offered to applications for the ontologies mapping. Instead in our approach, semantic conversations are used to reach an agreement over not shared concepts. However, a similar form of semantic agreement can be implemented in our framework as well: for instance, an ontologies-mapper Web Service can be easily integrated in the architecture by means of the publish/subscribe mechanism.

References

1. UDDI Spec TC. http://uddi.org/pubs/uddi-v3.0.2-20041019.htm.
2. P. Brusilovsky, S. Sosnovsky, and M. Yudelson. Ontology-based framework for user model interoperability in distributed learning environments. *Proceedings of World Conference on E-Learning*, pages 2851–2855, 2005.
3. F. Carmagnola. *From User Models to Interoperable User Models*. University of Turin, 2007.
4. F. Carmagnola, F. Cena, L. Console, O. Cortassa, M. Ferri, C. Gena, A. Goy, M. Parena, I. Torre, A. Toso, F. Vernero, and A. Vellar. icity - an adaptive social mobile guide for cultural events. In *In Proc. of Mobile Guide Workshop*, 2006.
5. F. Cena and L. Aroyo. A semantics-based dialogue for interoperability of user-adaptive systems in a ubiquitous environment. UM2007, Corfu, Greece, June 2007.
6. F. Cena, L. Console, C. Gena, A. Goy, G. Levi, S. Modeo, and I. Torre. Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AICommunication*, 19(4):301–400, 2006.
7. V. Chepegin, L. Aroyo, and P De Bra. Broker-based discovery service for user models in a multi-application context. In *ICALT '05*, July 2005.
8. V. Dimitrova. *Interactive Open Learner Modelling*. Leeds University, 2001.
9. Houben G.J. and K. van der Sluijs. Towards a generic user model component. UM2005, 2005.
10. L. Laera, V. Tamma, and J. Euzenat. Reaching agreement over ontology alignments. ISWC 2006, 2006.
11. J. Levin and J. Moore. Dialogue games: meta-communication structures for natural language interaction. *Cognitive Science*, 1(4):395–420, 1977.
12. J. Searle. What is a speech act. *Language and Social Context*, 1972.
13. M. Specht, A. Lorenz, and A. Zimmermann. Towards a Framework for Distributed User Modelling for Ubiquitous Computing. In *DASUM 2005 at UM05*, July 2005.
14. J. Vassileva. Distributed user modelling for universal information access. *Universal Access in Human - Computer Interaction*, 3:122–126, 2001.
15. D. Yimam and A. Kobsa. Centralization vs. decentralization issues in internet-based knowledge management systems:experiences from expert recommender systems. TWIST2000. Irvine, CA, 2000.

RSS-based Interoperability for Cross-System Recommendations

Yiwen Wang[1], Federica Cena[2], Francesca Carmagnola[2], Omar Cortassa[2], Cristina Gena[2], Natalia Stash[1], and Lora Aroyo[3]

[1] Eindhoven University of Technology, Computer Science
y.wang,n.v.stash@tue.nl
[2] University of Turin, Computer Science
cena,carmagnola,cortassa,gena@di.unito.it
[3] VU University Amsterdam, Computer Science
l.m.aroyo@cs.vu.nl

Abstract. This paper presents an approach to exploit widely used tag annotations to address two important issues in user-adaptive systems in the cultural heritage domain: the cold-start problem and the integration of distributed user profiles. The paper provides an example of re-use of user interaction data (tags) generated by one application into another one in similar domains for providing cross-system recommendations.

1 Introduction

The *Web 2.0* phenomenon, coined by O'Reilly Media in 2004 [10] introduced various social applications enabling online collaboration and encouraging the participation and contribution of spontaneous social networks. According to Yahoo! "the future of the web is about personalization...weaving the web together in a way that is smart and personalized for the use."[2]. User modeling that support personalization for a single application are reasonably well understood. However, contemporary users typically roam in a distributed information world, in which also a range of devices play a role when they interact with the information. Although the experiences become more engaging, more targeted and more effective across multiple applications, these are still "digital islands" in terms of personalized experience - not truly interconnected in a way that allows users to capitalize on the full potential of a distributed multi-application environment. The user data collected by different systems is fragmented over all of them and the user profile is inherently distributed and often not consistent.

The focus of this paper is to illustrate an approach to realize personalization in distributed and interactive environments. We show how existing fragments of user data in the form of tags can be brought together with the help of explicit semantics, and in this way allow for an adequate personalized experience across the boundaries of particular applications and devices. This poses a considerable number of technological demands. Working in a distributed setting implies that personalization has to consider both *data-integration issues* (how is information from different applications and devices related) as well as *context-modeling issues* (in which space/time/mode are statements about a user valid). In this paper we look at the data-integration issues, as a first step in reaching the end goal. Concretely, we provide a method for extracting, conceptualizing and linking user tags contained in public RSS files generated in the interaction of users with a social recommender system iCITY [3]. The tags are mapped to art-related concepts used in the personalized museum applications CHIP. Note that RSS files are offered by the majority of social applications and provide a real, even though a limited, solution to the interoperability challenge in distributed context.

One of the advantages of the approach we take is its potential for overcoming the "cold start problem" in user-adapted systems. It not only provides applications with user data even if they have not used the system yet, but it also does not load the user with yet another "interest filling in" activity. Additionally, both systems get a more exact view on the user's interests.

The paper is structured as follows: section 3 describes the functionality and architectural specifics of iCITY and CHIP systems. These are further elaborated in section 4 as interoperability aspects, such as conceptualization of iCITY tags and storing such tags in the CHIP user profile. Finally, in section 5 we draw conclusions and future work trends.

2 Related Work

Users are increasingly involved in multiple Web 2.0 environments, such as Facebook.com, Flickr.com, Del.icio.us, etc. Most of those services maintain a different identity, e.g. login information, preferences or profile of users but there is a *limited integration of these user profiles* between different applications and still not a methodological approach of how to assess the users experience and improve in an *evolutionary* way the provision of the services.

Social tagging is of utmost relevance to the Cultural Heritage domain, which may be of help: (i) to bridge the gap between the professional language defined by domain experts and the popular "un-trained" language; (ii) to encourage individuals to find personal meanings/perspectives in public collections by labeling the cultural heritage resources (e.g. artworks); and (iii) to create public engagement with cultural heritage collections. Projects that explore this challenge, such as the Steve Museum [12], demonstrated the effectiveness of social tagging in engaging visitors and for the museum to understand what users consider as relevant. The Powerhouse Museum [6], proved that user tagging and folksonomies can improve navigation and discoverability through the museum collection.

Furthermore, user tags could be extremely useful for adaptive web applications [9], e.g. to enrich and extend the user model [4]. Annotations can become part of the user profile as "an indication of his perspective on the content collection and interest in the annotated object" [13], thus, the systems can obtain from the tags the user has inserted, knowledge about preferences, interests, etc. Adaptive systems may use this "tag-enriched" profile for recommendations. No-

tice that tagging, and more generally annotating, can be considered as possible actions a user can perform on a social web site. As other kinds of *usage data* [8] (clicking, buying, etc.) these actions represent an important feedback from the user. In fact user usually tags to highlight and organize the items she is interested in, in order to retrieve them later. Thus the action of tagging is a stronger indicator [8] for user interests than simply clicking on a link, and therefore should be be analyzed in order to make interesting inferences on the user model.
To be able to exploit tags for improving the user model, systems are required to understand the semantics of the tags. Suitable strategies borrowed from automatic Word Senses Disambiguation (WSD) are applied. This involves matching the context of a word instance with either information from an external knowledge source (knowledge-driven WSD), or information about the contexts of previously disambiguated instances derived from corpora (data-driven or corpus-based WSD) [7].

3 iCITY - CHIP User Interoperability Architecture

The section illustrates the main characteristics of iCITY and CHIP user-adaptive systems and the interoperability aspects of their architectures (Fig. 1).

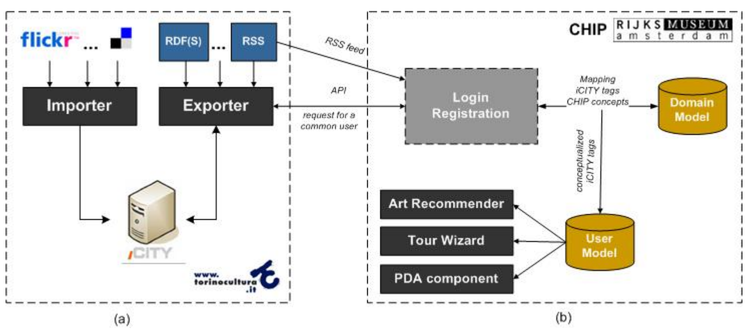


Fig. 1. iCITY-CHIP Interoperability Architecture

3.1 iCITY Tagging and Recommender System

iCITY is a social web-based, multi-device recommender system. It provides suggestions on cultural events in the city of Turin, and allows users to insert new events, to add information about events, to insert comments and tags. Recommendations are based on the user model enriched with tags, exploited to infer user features (see [4]). The iCITY user model is an overlay of the iCITY Event ontology, created as an RDFS transformation of the event classification in

TorinoCultura[4], a web portal managed by the municipality of Torino for informing citizens about cultural ongoing events in the city. This ontology contains links both to WordNet[5] synsets and domains. iCITY has a modular architecture for extracting, maintaining, reasoning and exporting of user tags (Fig. 1(a)), which can be shared with other applications via a RSS feed. The main components for interoperability (Fig. 1(a)) are the Importer and Exporter Modules responsible respectively for the extracting the tags from external sources and making user tag available to other applications.

iCITY Importer Module. In the iCITY registration form, a part from username and password, the user can provide the tags that best describe her and her social web community. If the user provides web community accounts (e.g. flickr.com, del.icio.us), the Importer Module retrieves the available RSS files containing the set of tags provided by the user in those web communities [5]. The Importer Module is able to extract these tags regardless of the format used to represent them. Once all the user tags have been extracted, they could give interesting information about user interest and knowledge. In order to understand their meaning, the system looks for correspondences between the tags and the *synsets* and the *domains* of the MultiWordnet database[6]

If one or more correspondences are found, they are linked to the Event Ontology class/subclass. Thus, the system can reason on tags and increases the level of inferred user's interests related to the class the tag belongs to. However, this approach is limited and suffers from several problems. Whether the tags could not be directly mapped on the concepts in iCITY ontology iCITY loses some important information.

iCITY Exporter Module. The Exporter Module generates, for every user, a RSS file with the list of the events tagged by the user. For every event, the file stores: the title, the URL , the description, the reference to the event category and subcategory in the iCITY event ontology, the reference to the Wordnet synsets and domains linked to the subcategory, and finally the list of the tags associated by the user to the event. Notice that, the list of tags ($< dc : subject >$) is expressed as a simple list of meaningless strings.

The only way to provide semantics to them for the receiver is exploiting the reference to the category and subcategory the event belongs to in the iCITY event ontology, and the reference to the WordNet domains and synsets[7]. In this way, a recipient system can import the RSS file containing the tags used by a particular user, and it can try to disambiguate the meaning of the tagged event

[4] http://www.torinocultura.it/

[5] http://wordnet.princeton.edu/

[6] In MultiWordnet (http://multiwordnet.itc.it/), and Wordnet, each synset is annotated with at least one domain label, selected from a set of about two hundred labels which constitute the so-called WordNet Domains.

[7] We provide two different synset IDs for every sysnet: the former is referred to the database location id shown in the online version of WordNet (http://wordnet.princeton.edu/perl/webwn); the latter refers to the ID given to the synsets in MultiWordet

thanks to the information, provided in the RSS file, about the event subclass they belong to and the references to WordNet domain and synset.

In the example, after tag disambiguation, the receiver could infer from the tags a user interest in Art, and in particular contemporary art, or compute the level of active participation of the user in the system depending on how many tag the user inserts [4].

3.2 CHIP Artwork Recommender and Personalized Tour Guide

CHIP system (Cultural Heritage Information Personalization) allows users to create their personalized museum experiences both with the virtual collection on the Rijksmuseum Web site and in the physical museum space by using a PDA mobile device. Our goal is to demonstrate how Semantic Web and user modeling technologies can be deployed to enhance the personalized museum access.

For the semantic enrichment of the Rijksmuseum ARIA[8] digital database (750 master pieces), we used mapping of ARIA terminology to external vocabularies and thesauri [1], namely the three Getty vocabularies[9]: Art and Architecture Thesaurus (AAT), Union List of Artists Names (ULAN) and the Thesaurus of Geographical Names (TGN), as well as the subject classification Iconclass[10]. The use of common vocabularies provides the new data repository a relational and hierarchical structure for reasoning and making recommendations.

Based on the semantic-enriched data model, we implemented three different tools/components in the CHIP demonstrator: (i) Artwork Recommender, (ii) web-based Tour Wizard, and (iii) PDA-based Mobile Tour.

- *Artwork Recommender*, a Web-based rating dialog for artworks/topics to build a user profile, based on semantics-driven recommendations.
- *Tour Wizard*, a Web-based tool using the user profile to generate (semi)automatically personalization virtual museum tours for each user.
- *Mobile Tour*, a PDA-based tool to map virtual tours into the physical museum space with constraints (e.g. available artworks, time); to give guidance to users and to synchronize the user profile on the web and in the PDA.

To collect various user interactions from these different tools, we took an open Web and 'non-intrusive' approach to build a user profile with four different categories of user information: *personal*, *social*, *iteraction* and *ratings*. (i) *personal* category stores the user stable characteristics (e.g. name, age, gender, mbox), which in a typical open Web context could be initialized by either importing an existing FOAF RDF profile or via an OpenID channel linking the CHIP login data to an existing login information of third party Web application; (ii) *social* category describes the user's social information also initialized by FOAF properties,e.g knows, openid, organization, OnlineAccount, etc; (iii) *interaction* category stores the user's various interactions from different CHIP

[8] http://rijksmuseum.nl/aria/
[9] http://www.getty.edu/research/conducting research/vocabularies
[10] http://www.iconclass.nl/libertas/ic?style=index.xsl

tools, e.g. virtualTours (artworks created in the Tour Wizard on the Web), realTours (artworks visited in the real museum), time duration in the real tour, etc.; (iv) *ratings* is the core category, which maintains the user's explicit ratings of artworks and topics in terms of VRA Core properties[11], e.g. work, creator, title, creationDate, creationSite, subject, etc.

4 iCITY-CHIP User Tag Interoperability

In this interoperability use case we use an open API to request and link the user data. Once the user personal (login) information is aligned between CHIP and iCITY (Fig.1(b)), based on the RSS feed we maintain a dynamic mapping of iCITY user tags to the CHIP vocabulaty set: Rijksmuseum specific ARIA concepts, shared domain vocabularies (e.g. Getty AAT, ULAN, TGN, and IconClass) and general purpose lexical data such as WordNet.

The main challenge in achieving the interoperability of user tags between the two systems iCITY and CHIP is to provide a dynamic mapping mechanism, which allows for a constant stream of user tags from iCITY to be interpreted (mapped) to concepts from the internal vocabularies of CHIP. This will allow to use iCITY tags to populate the user profiles of users (especially first-time users) in CHIP and to be able to instantly generate recommendations in the Rijksmuseum collection.

To implement the tag mapping from iCity to CHIP, we use the Simple knowledge organization System (SKOS) Core Mapping Vocabulary Specification[12] created for linking thesauri to each other with relationships, e.g. skos:equivalentConcept, skos:broader, skos:narrower and skos:related. For the first stage alignment, the mappings are still based on the lexical match of tags. With a few additional simple restrictions by applying the type of tags, a lexical match gives more confidence to generate a strong semantic match [11], e.g. a semantic equivalence mapping to *"Amsterdam"*, known to be the geographical name of a city, can be made with owl:sameAs[13]. The mapping from iCity user tags *"Amsterdam"*, *"Giovanni"*, *"photography"* and *"1400"* to CHIP vocabularies is realized in two steps: (i) to identify the type (e.g. creator, place, material, etc.) of tags as a simple restriction; (ii) to map tags by using SKOS Core mapping relations. The iCITY tag *"Giovanni"* results in two partial matches in the Getty Unified List of Artist Names (ULAN) with the type of `creator`: *"Piranesi, Giovanni Battista"* and *"Tiepolo, Giovanni Battista"*. The level of ambiguity with names as user tags could be quite high, especially if the domain in not only limited to art. To confirm whether *"Giovanni"* is either of the two Getty ULAN artists or none of these two, we need further evidence from the user tag cloud, e.g. the event (annotated with this tag) and related tags (used together to annotate this event). From the user's iCity RSS file, we know that *"Giovanni"* is used to annotate the event *"Why Africa?"* together with other tags *"Africa, exhibition,*

[11] http://www.vraweb.org/projects/vracore4/index.html
[12] http://www.w3.org/2006/07/SWD/wiki/SkosDesign/ConceptualMapping/ProposalOne
[13] http://www.w3.org/TR/owl-features/

art, contemporary, Torino" and actually the user means *"Giovanni Agnelli"*. Thus in this case, although we have good partial match, we use the semantic weak "`skos:related`" mapping relationship with a low certainty. Another example, the semantic equivalence between iCITY tag *"Amsterdam"* and Getty Thesaurus of Geographic Names (TGN) creationSite *"Amsterdam"* is expressed with skos:equivalentConcept for the type of `place`.

Compared with *"Giovanni"* (2 weak related matches), *"Amsterdam"* (1 exact match), and *"1400"* (no matches), the mapping of *"photography"* is more complex, which results in 8 partial matches in the CHIP vocabularies with different SKOS mapping relations and different tag types: (i) the `skos:narrower` for the type `material` points to *"Photo/collotype"*, *"Photo/Gelatin silver print"* and *"Photo/Bromide print"* in the hierarchical specialization of Rijksmuseum ARIA vocabulary; (ii) the `skos:broader` for the type `subject` indicates *"Photographs"*, *"Photographic techniques"*, *"Photography as inspiration"* and *"Photography books"* in ARIA as well; (iii) the `skos:equivalentConcept` for the type `subject` refers to the IconClass concept vocabulary *"Photography"*, which describes a semantic equivalence; and (iv) again, a semantically weak relationship `skos:related` for the type `artwork` is applied for *"Jerusalem, etude et reproduction photographique des monuments de la Ville Sainte"* in ARIA.

These examples above give a good illustration of the semantic and syntactic mappings we can provide between the iCITY tag cloud and the CHIP art concepts. Maintaining a certainty level for each mapping allows for tuning of those concept's relevance. Further evidences from the tag cloud and/or the user model allows for a better accuracy as well as the user's direct feedback/confirmation. However, there are still some problems remaining. *i)* Tags can be messy. The mapping is realized in two steps: first, to identify the type of the tag, e.g. whether *Giovanni* is a person, a kind of material or a place; and then to the map tags using SKOS mapping relations. *ii)* Grammatical variation. Often tags appear in various grammatical forms, which do not completely match the CHIP concept form, e.g. noun, verb, adjective and adverb formas. Maintaining additional relationships or distances between the different term forms allows for clustering of all possible mappings for a given tag, e.g. sculpture - sculptural - sculptor, theater - theatrical. Using mapping to WordNet can facilitate this process efficiently. *iii)* Combined effect of tags for recommendations. After the tag mapping, we face the issue of generating recommendations in CHIP based on the mapped tags. Considering tags are often used together, we should treat the tags differently depending on their relations with the annotated events, which are described in the user's RSS file. *iv)* Ranks of recommendations. To rank the recommendations based on different tags or tag groups, we are considering maintaining a dynamic weight for each tag, which could be defined by some factors, e.g. *frequency*, *uniqueness*, etc.

5 Conclusion and future work

In this paper we have presented an approach to exploit widely used tag annotations to address two important issues in user-adaptive systems in the cultural heritage domain: the cold-start problem and the integration of distributed user profiles. We have sketched a scenario, in which user tagging about cultural events gathered by iCITY is used to enrich the user profile for generating personalized recommendations of artworks in CHIP. To realize full tagging interoperability, we have investigated the problems that arise in mapping user tags to various ontologies , and we proposed additional mechanisms, such as the use of SKOS matching operators, to deal with the possible mis-alignment of tags and domain-specific ontologies. Issues that need to be addressed in future research are the loss of information that occurs when relating tags to event ontologies (iCITY) and the effective mapping between of single or possibly multiple tags to the domain-specific ontologies as used in cultural heritage.

Last but not least, the CHIP user model has to be exported back into iCITY recommender in a interoperable format, in a way that the iCITY Importer module should be able to import this information. The inferences made by the CHIP recommender could be extremely useful in order to refine the iCITY user model and could also help the iCITY reasoning component to solve some of the disambiguation problem described in 3.1. On the other side, the CHIP mapping component could be refined in order to consider also the sysnets and the domains information exported in the iCITY RSS file to have an help in solving its disambiguation problems.

References

1. L. Aroyo, N. Stash, Y. Wang, P. Gorgels, and L. Rutledge. Chip demonstrator: Semantics-driven recommendations and museum tour generation. In *ISWC/ASWC*, pages 879–886, 2007.
2. T. Bhat. ”search is history”, says yahoo! 2007.
3. F. Carmagnola, F. Cena, L. Console, O. Cortassa, M. Ferri, C. Gena, A. Goy, M. Parena, I. Torre, A. Toso, F. Vernero, and A. Vellar. icity an adaptive social mobile guide for cultural events. *Proc. of Mobile Guide 2006*, 2006.
4. F. Carmagnola, F. Cena, O. Cortassa, C. Gena, and I. Torre. Towards a tag-based user model: how can user model benefit from tags? *Proc. of UM2007, LNCS*, 4511:445–449, 2007.
5. F. Carmagnola, F. Cena, and C. Gena. User modeling in the social web. In B. Apolloni, R.J. Howlett, and L.C. Jain, editors, *KES (3)*, volume 4694 of *Lecture Notes in Computer Science*, pages 745–752. Springer, 2007.
6. S. Chan. Tagging and searching: serendipity and museum collection databases. In *Museums and the Web 2007*, pages 87–99, 2007.
7. N. Ide and J. Veronis. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40, 1998.
8. A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2):111–155, 2001.

9. M. Maybury and P. Brusilovsky. *The Adaptive Web*, volume 45. Communications of the ACM, 2002.
10. T. O'Reilly. What is web 2.0. design patterns and business models for the next generation of software. 30/09/05.
11. A. Tordai, B. Omelayenko, and G. Schreiber. Thesaurus and metadata alignment for a semantic e-culture application. Semantic Authoring, Annotation and Knowledge Markup Workshop, 2007.
12. J. Trant and B. Wyman. Investigating social tagging and folksonomy in art museums with steve.museum. In *World Wide Web (WWW '06)*, 2006.
13. M. van Setten, R. Brussee, H. van Vliet, L. Gazendam, Y. van Houten, and M. Veenstra. On the importance of "who tagged what". Workshop on Social Navigation and Community-Based Adaptation Technologies, 2006.

Towards Cross-System User Model Interoperability for Planning Lifelong Learning

Hassan Baajour[1],George D. Magoulas[1,2], Alex Poulovassilis[1,2]

[1] London Knowledge Lab, Emerald Street, London WC1N 3QS
[2] Department of Computer Science and Information Systems,
Birkbeck College, University of London, WC1E 7HX
{hassan, gmagoulas, ap}@dcs.bbk.ac.uk

Abstract. User model interoperability offers opportunities for acquiring relevant fragmented information about learners in order to provide cross-system personalisation and support for planning lifelong learning and related activities throughout life. To this end, in this paper, we adopt a service-based approach and develop a comprehensive ontology for planning lifelong learning that could serve as a reference model in order to support the development, mapping and integration of user models in the context of a lifelong learning network.

Keywords: Ontology-based user modelling, interoperability, lifelong learning networks.

1 Introduction

In Lifelong Learning Networks (LLNs) infrastructures should be able to support cross-system personalisation by exploiting user models embedded in various systems. [1] [2]. This paper illustrates the importance of user model (UM) interoperability in order to enable learner information that has been accumulated in one learning context of a LLN to be reused in another in as highly an automated fashion as possible and operates within the E-Learning Framework-ELF (http://www.elframework.org/). The basis of our approach is an ontology-based user model, presented in Section 2. In Section 3, we discuss the role of the developed ontology in an LLN, and describe a service-based broker approach for cross-system personalisation taking an example of user model interoperability from the MyPlan project[1]. Section 4 presents conclusions.

2 A User Ontology for Planning Lifelong Learning

The development of comprehensive ontology-based user modelling requires particular support in its design and structure in order to elicit and organise the user models in a consistent, reusable and transparent fashion [3]. To this end, there have been several

[1] The MyPlan- Personal Planning for Learning through Life project (http://www.lkl.ac.uk/research/myplan/) is funded by the e-Learning Capital Programme of the Joint Information System Committee, UK.

attempts to create user ontologies based on a variety of methods in recent years. Examples are the OntoClean (ontoclean.org) and IDEF5 (idef.com/IDEF5.html) methodologies, and the Onto-logging [4] and the Elena projects [5]. Also [6] introduced the general user model ontology (GUMO) for the uniform interpretation and communication of distributed user models in intelligent semantic web enriched environments. Lastly, [7] proposed the Shared User Model (S-UM) for mapping various user models to each other in multiple educational applications by a two-way mapping for each schema to and from every application and the shared user model.

Our work adopts Noy and McGuinness's methodology [8], which combines standards and general information dimensions, as in the above mentioned works, and breaks down the process of building an ontology into seven basic steps: determine the scope, consider reuse of existing ontologies, enumerate important terms, define the classes and their hierarchy, define the properties, define the facets of the slots and create instances. The proposed ontology aims to capture and enumerate rich metadata that cover the main activities performed by users in planning their lifelong learning and was built based on the following:

(i) An analysis of lifelong learners' requirements conducted during the first phase of the L4*All* project (http://www.lkl.ac.uk/research/l4all/). Additional consultations have been conducted within the MyPlan project advisory group and with experts in lifelong learning, including workshops, face-to-face and telephone interviews with specialists in the field and a scoping study of other projects for planning lifelong learning in UK and services in the London area.

(ii) A review of the most relevant metadata standards relating to learner profiles, such as *FOAF, IMS LIP, vCard, UKLEAP, EduPerson, PAPI, UKgov,* that was informed by the regional dimension of our project.

(iii) A review of several open source lifelong learning systems, such as the L4*All*, *Passportfolio* (www.passportfolio.com/), e-Portfolio for Lifelong Learning Reference Model "*eP4L"* (www.nottingham.ac.uk/epreferencemodel/), and *Epsilen* (www.epsilen.com).

Twenty five concepts were identified in the planning of lifelong learning and included the Lifelong Learner Ontology (LLO), which has been implemented using Protégé (http://protege.stanford.edu/). Central to our approach is the *Timeline* concept conceived by the L4*All* and MyPlan projects, and its related classes and properties. Creating timelines allows users to plan their lifelong learning in the short and long terms, and to reflect upon their educational and career choices. This can help lifelong learners to obtain a vision of how to organise and structure their lifelong learning and work activities. The "Timeline" concept includes episodes that specify activities attended or future plans. Episodes can be categorised into three major groups, namely "Educational Episodes", "Occupational Episodes" and "Personal Episodes", and can have start date, end date, name, type, description, comment and ID. Properties that represent relationships between different concepts are also defined; for example the property "isTimelineOf" links a particular timeline to a particular lifelong learner. A Timeline concept owns the 'hasPrivacy' property which links the Timeline with the Privacy concept/class that contains data relevant to the learner's privacy issues (e.g. learners may or may not give permission for releasing their personal data for a particular purpose such as a personalisation service and can specify whether or not they allow the data to be transmitted to a third party). The Timeline allows monitoring learner's current activities and/or future plans and can be used to regularly update Lifelong Learner model related concepts/classes, such as Activity, Accessibility, Qualification, Affiliation, Competence, Interest, Goal[2] and Product with the proper information; this can de further used to support personalisation.

3. The LLO and an Architecture for User Model Interoperability

In line with previous work in the area of user model interoperability [7], the LLO is intended to be used as reference model for merging and mapping the various user model schemas arising in the planning of lifelong learning. To efficiently manage distributed UMs in this context, an architecture for UMs exchange and management is needed [10]. In this direction, there is a general trend towards designing services and ontology standards-based architectures to enable effective educational systems interoperability [11] [12] [13] [14].

Our work aims to support interoperability between systems that adopt the ELF within a LLN. Therefore we have chosen to use a broker service-based architecture for accessing, mapping, transferring and integrating UMs to allow personalised applications to expose and share available UMs existing in a LLN. The architecture shown in Figure 1 illustrates how a service-based framework and the reference model ontology will facilitate UMs interoperability between two systems: the L4*All* and eProfile (www.essex.ac.uk/chimera/projects/eProfile.html). This architecture consists of: (i) a Web Portal that provides a graphical user interface that represents the main gate to the LLN. It allows LLN users to access various services provided by different systems within the LLN and includes web components that activate collaboration between various services to support the lifelong learner activities within the LLN; (ii) a Broker that includes services to exchange user information, to update user models, to merge and map the different user model ontologies and schemas according to the LLO. This architecture provides UM interoperability at two levels:

(i) the syntactic level, where the broker provides components and services for accessing and exchanging user information as needed. This level relates to the transfer, storage and retrieval of user-related information, using standard web-services technologies such as UDDI, WSDL and SOAP.
(ii) the semantic level, which enables the interpretation and integration of user-related information from various systems. In this context a semantic data integration tools, such as AutoMed[3] [9] heterogeneous data integration toolkit, is needed.

In Figure 1, the Broker component contains services that are not specific to a particular system, which are used to invoke, query and consume the UM information via the AutoMed data integration component. The broker also includes a UM updating agent, which is used to check the information available about a particular

[2] Theses seven concepts are main classes in the IMS-LIP metadata model http://www.imsglobal.org/profiles/lipinfo01.html

[3] Automed transforms each schema into a Common Data Model (CDM), chosen by the integrator in order to remove the heterogeneity of the various schemas' data models. All source, intermediate and integrated schemas, and the mappings between them, are stored in AutoMed's Schemas Transformations Repository.

user in the LLN and henceforth update the UMs of other systems with the relevant fragments of the most recent user data, if needed. AutoMed can be used as a mediation service between the two systems, where it is invoked by a service (see Broker-No4 in Fig. 1), receives its output, transforms it into the proper format and makes it ready for consumption by another service (see Broker-No5 in Fig. 1).

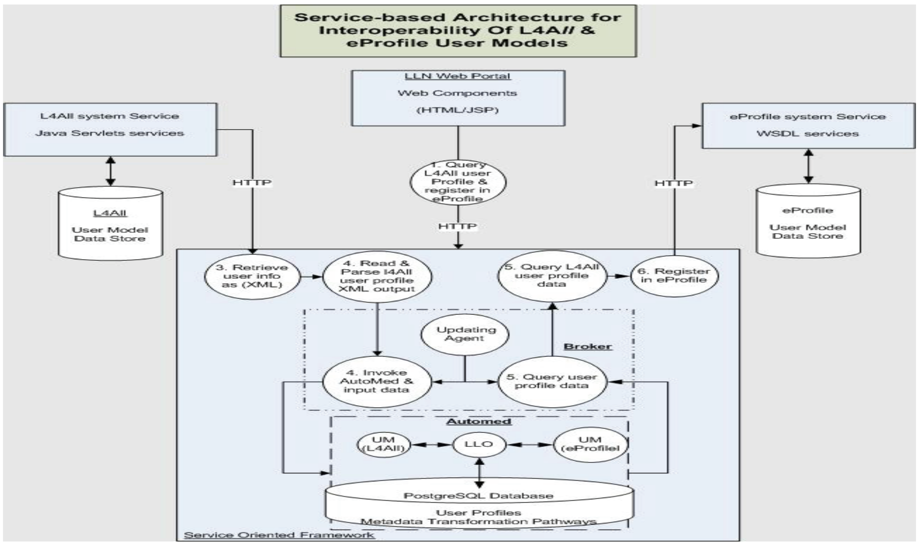


Fig. 1 Service-based broker architecture to enable UM interoperability.

4. Conclusions

This work focused on investigating UM interoperability for the planning of lifelong learning in the context of a Lifelong Learning Network. The first aspect of the research concerned creating an ontology-based user model whilst the second aspect concerned proposing a service-oriented broker-based architecture to facilitate communication between systems and mapping of different user model schemas according to the Lifelong Learner Ontology developed.

An interesting extension of our work will be going into semi- or fully automatic solutions for mapping user models into the LLO. Automatic mapping between different personalised systems' user models into the LLO will enable more effective interoperability between them. In addition, there is a need to tackle the many security and privacy concerns from a technical perspective in order to protect users' privacy rights and to encourage them to use the infrastructure of the Lifelong Learning Network in a safe and secure manner.

References

1. Brouns, F., Bitter-Rijpkema, M.E., Sloep, P.B., Kester, L., van Rosmalen, P., Berlanga, A.J., Koper, R. Personal profiling to stimulate participation in learning networks, ePortfolio Conference (2007) available online http://www.eife-l.org/publications/eportfolio/proceedings/ep2007/papers/eportfolio/profiling-mechanisms-to-stimulate-participation-in-learning-networks
2. Koper R. Building Learning Networks for Lifelong Learning, (2007) available online at: http://dspace.ou.nl/bitstream/1820/1077/5/DiesAddress-Koper-text-only.pdf
3. Denaux, R., Aroyo, L., Dimitrova, V. An approach for ontology-based elicitation of user models to enable personalization on the semantic Web, 14th International World Wide Web Conference, Chiba, Japan. pp. 1170-1171 (2005) Available online at: http://www2005.org/cdrom/docs/p1170.pdf
4. Razmerita, L., Angehrn, A., Maedche, A., Ontology based user modeling for knowledge management systems. In Proceedings of the 9th International Conference on User Modeling, Pittsburgh, USA, Springer-Verlag, pp. 213-217 (2003)
5. Dolog, P., Nejdl, W. Challenges and benefits of the semantic Web for user modelling. In Proc. of AH2003 Workshop at 12th WWW Conference, Budapest, Hungary (2003) available online at http://citeseer.ist.psu.edu/dolog03challenges.html
6. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., Wilamowitz-Moellendorff, M.. GUMO - The General User Model Ontology. In: Ardissono, L., Brna, P., Mitrovic, A. (Eds.), Proc. of the 10th International Conference on User Modelling, July 2005, Edinburgh, UK. pp. 428-437. (2005)
7. van der Sluijs, K. and Houben, G-J. A generic component for exchanging user models between web-based systems, Int. J. Continuing Engineering Education and Lifelong Learning, Vol. 16, Nos. 1/2, pp.64–76. (2006)
8. Noy N.F. and McGuinness D. L. Ontology Development 101: A Guide to Creating Your First Ontology'. Stanford, CA, Stanford University. (2001) Available online at http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf
9. McBrien P.J. and Poulovassilis A., Data Integration by Bi-Directional Schema Transformation Rules, Proceedings ICDE'03, Bangalore, pp 227-238, IEEE Computer Society, March (2003)
10. Aroyo, L., Dolog, P., Houben, G-J., Kravcik, M., Naeve, A., Nilsson, M., Wild, F.. Interoperability in Personalized Adaptive Learning. Educational Technology & Society, 9 (2), 4-18 (2006) http://www.ifets.info/journals/9_2/2.pdf
11. Devedzic, V. Next-generation web-based education, International Journal for Continuing Engineering Education and Lifelong Learning, Vol. 11, Nos. 1/2, pp.232–247. (2003)
12. Dicheva, D. and Aroyo, L. An approach to interoperability of ontology-based educational repositories, Int. J. Continuing Engineering Education and Lifelong Learning, Vol. 16, Nos. 1/2, pp.92–109 (2006)
13. Meccawy M., Celik I, Cristea A. I, Craig, Stewart D., Ashman H. Interoperable Adaptive Educational Hypermedia: A Web Service Definition. Proc. of ICALT pp: 639-641 (2006)
14. Celik I, Stewart C, and Ashman H, Interoperability as an Aid to Authoring: Accessing User Models in Multiple AEH Systems, in Proc. 4th Adaptive and Adaptable Educational Hypermedia workshop at the Adaptive Hypermedia (AH) 2006 conference, (2006)

An Interoperability System for User Models in Adaptive Educational Hypermedia

Ilknur Celik[1], Helen Ashman[2], Peter Blanchfield[1], Tim Brailsford[1]

[1] School of Computer Science & IT, University of Nottingham, Wollaton Road, Nottingham, NG8 1BB, UK
{ixc, pxb, tjb}@cs.nott.ac.uk
[2] School of Computer and Information Science, University of South Australia, GPO Box 2471, Adelaide, South Australia 5001, Australia
Helen.Ashman@unisa.edu.au

Abstract. This paper introduces an interoperability framework for user models between Adaptive Educational Hypermedia (AEH) systems. The framework classifies the user model attributes into four groups: Identifiers, Adaptation, Presentation and Domain. Mapping between user models is based initially on the attributes of the user model and then refined through the information contained in the user profiles. Experimental results are encouraging with high levels of success in mapping variables between the different AEH systems used in the tests. The paper concludes with a discussion of the future work which will involve ways of mapping interdependencies of the user model attributes.

Keywords: User Model, Interoperability, Adaptive Educational Hypermedia, Adaptive Hypermedia, User Profile, Personalization, MOT, WHURLE, DEUS.

1 Introduction

Creating good quality AEH [2] systems is a time-consuming, expensive and complex process, which in practice almost never occurs outside of research labs. Considering the difficulties in authoring and maintenance of AEH systems (to a great extent due to a dearth of tools), the idea of interoperability between AEH systems becomes not only desirable but also necessary [3]. The aim of interoperability is to allow dynamic interchange of data between AEH systems, i.e. the re-use of previously created material and the automated transfer of user information, which eliminates the cost of re-authoring content and the loss of valuable user data acquired over a long time.

The information that is used by AEH systems in order to determine the nature and scope of adaptation is derived from user profiles. Therefore, when attempting to port content between disparate systems if there is no transfer of users' individual information between these systems, then the adaptive behavior itself cannot be ported. For instance, when an AEH author creates content in one system, and exports this along with the user profiles to other AEH systems, the students should be able to access that material in any system of their choice. When students access a particular lesson, the information presented to them has to be adapted according to their user profile for the relevant user model (UM). Based on the interactions there, their user profiles should be updated accordingly. Thus, the next time the same user employs a different AEH system, the user would be presented with adaptive content according to the updated user profile. In order to achieve this "many-to-many" paradigm [15] in interoperability of AEH systems, interoperability of content is insufficient. UMs must also be interoperable between the AEH systems.

The aim of our research is to enable interoperability between AEH systems to some extent. This requires interoperability of materials, adaptation rules, UMs and presentation [15]. Previously, we transferred the user profiles along with the learning material, and attached lessons from MOT [5] to a WHURLE-compatible format, which enabled, for instance, a teacher to transfer the content authored in MOT to create lessons in WHURLE-HM [16], thus we provided interoperability of materials and UMs using the conversion approach [14]. Examining this and the previous experiments with other conversions such as MOT to AHA! [6] and MOT to Blackboard [12], we extracted common patterns that should promote interoperability between other AEH systems. The intention is to move away from the 'one-to-many' towards a true 'many-to-many' paradigm.

Our next step in achieving this more general interoperability particularly for UMs is to implement an interoperability system that reflects the existence of interdependencies between the components that make up an AEH system, such as the UM, adaptation, presentation and domain. For instance, an AEH system adapts the presentation of a given domain depending on the current information from the user profile and applying the adaptation rules to this information. In our methodology, we make use of these interdependencies to classify the UM attributes depending on their functionality in the adaptive environment. Our initial work has begun the implementation of this and demonstrated its operation for a number of AEH systems.

Our research can draw parallels to the recent tendency in using Semantic Web technologies (i.e. decentralized UMs [8], user modeling services [4], schema conversion [13], ontology based user modeling [9], and standards [10] as in [7]) in order to improve interoperability of user profiles.

2 Mapping User Models between AEH Systems

We classify the UM attributes (variables) into the four main categories: Identifier, Adaptation, Domain and Presentation, which helps in identifying the relationship between the UM attributes from different systems, making the whole translation task manageable. The entities to map for each system are the UM attributes, the adaptation rules, the presentation rules and the domain rules. The UM variables are automatically classified into one of these categories based on an extensible translation algorithm. The UM attributes that do not fall into any of these categories are stored separately as they are considered to be out-of-vocabulary (OOV) entities. The OOV entities are stored to enable backward translation without loss of data.

The necessary information required for the translation process is extracted based on the database information (i.e. list of variables, access information, database structure such as table names, etc.) provided during the registration process. The variable-only translation algorithm (described below) is applied to classify the UM variables into one of the four categories discussed, or else as OOV. Finally the interdependency translation algorithm (under development) is applied to the data to show which variables are common, common-like and OOV in the other systems with regard to the interdependencies in order to provide practical interoperability.

2.1 The Variable-only Translation Algorithm

The algorithm for translating the variables looks at the UM variable names, frequency, values and types in order to classify them into one of the following categories; Identifier, Adaptation, Presentation or Domain. The algorithm first translates the variables by classifying them without the adaptation rules.

The algorithm initially looks at variable names for generic words or expressions that would help in classifying them based on the patterns revealed from our previous research [3]. Example expressions are: *address, lesson, knowledge* and *preference*. Each variable with an identified expression in its name is given an appropriate weight and stored in the relevant category for further processing e.g. address in Identifier, lesson in Domain, knowledge in Adaptation, and preference in Presentation.

The algorithm then looks at the variable values (from the user profiles) for numbers, words, sentences, dates or empty values. This classifies them according to types in order to study the values. If a variable's values are empty and its name does not include one of the general terms, the algorithm classifies it as OOV. The empty-valued variables are considered optional, and not significant for adaptation.

Next the variable values for numbers are analyzed by looking at their uniqueness, frequency, range or scale to put them into one of the four categories.

The algorithm then looks for date type variables, using different formats of representing dates (e.g. numeric, USA style, etc.). The current date and time are compared with the variable value to draw a conclusion. For instance, a future date cannot be the date of birth of a student; rather they are assumed to be deadlines and therefore in some way related to the Domain, i.e. a lesson closing date, a test date, etc.

Lastly, the algorithm looks at the variable values for strings, either single words or sentences. The algorithm assumes that variable values consisting of more than five words have a high probability of belonging to the Domain category. The algorithm also looks for specific words/expressions/terms/values in the variable values such as *beg, int, adv, expert, novice, yes, no, female, male,* etc. and adds highest weights to the corresponding variables and puts them in the appropriate categories.

The algorithm also analyzes the variables that appear in more than one category by examining the frequency or uniqueness of a variable's value to weigh them. It then calculates the weights for these variables to decide to which category they belong.

The algorithm is implemented using Perl in such a way that it can function with any given relational database, making it extensible and flexible. Firstly, the variable-only translation algorithm is applied to the map of UM variables to classify them into the four categories discussed above. These are then classified into common, common-like or OOV. Common and common-like are the variables that have the potential of having a similar functionality in another system, while the OOV variables are the ones that do not support a corresponding peculiarity of adaptation in other systems.

2.2 Test Results and Analysis

So far the algorithm has been tested with three different AEH systems. First we tested our algorithm with MOT and WHURLE-HM systems to compare the efficiency and the effectiveness of our methodology with our previous work [3] [14]. When we tested the algorithm with the online MOT UM [11], the algorithm successfully classified 23 out of 25 MOT UM variables (92%) into the four categories discussed. The only UM variables (*figures* and *machine-type*) that could not be classified were Presentation variables. In the MOT system, these are part of the optional preferences given to the users; hence they do not play a significant role in terms of adaptation in general. The algorithm could not classify these two variables because, being optional, they lacked values for most users. Next, the algorithm was tested with the WHURLE-HM UM where it effectively classified 23 out of 26 of the parameters (88%) into the discussed categories. The three variables that the algorithm could not classify were *categ*, *lrange* and *urange*. When they were analyzed, we found that these variables were static and did not affect the adaptation.

In our previous work, when we employed the conversion approach to transfer MOT UM along with the MOT content into WHURLE-HM, we achieved a series of adaptive lessons that could be viewed both in WHURLE-HM and MOT by MOT users. Nevertheless, these lessons had restrictions because of the problems with the number of UM attributes transferred between the two systems, such as loss of some variable values and inconsistencies in two-way transfers. However, with the translation algorithm approach, we managed to classify most of the UM variables in both systems. In total, with the translation algorithm the number of OOV variables is reduced from 29 to 5, compared to our previous work with the conversion approach.

Lastly, we tested our algorithm with the UM of the DEUS [1] system where it successfully classified 12 out of the 13 UM variables into one of the Identifier, Domain, Adaptation or Presentation categories, which is again 92% successful. *Lastdate* was the only variable that the algorithm could not classify due to all the values being set to NULL. Therefore, the algorithm classified this variable as optional (or OOV) as the NULL values indicate that it is not actually being used by the system. This variable is presumed not to have an impact on the adaptation of the system; as a result it will not influence the mapping of the variables across multiple systems.

3 Conclusion and Future Work

We found from our previous work that translating UM variables alone is not sufficient for interoperability of AEH systems. This is especially true for what we called the OOV variables. This implies that the interdependencies in one system’s variables will also need to be translated into the other system. The experiments with WHURLE, MOT and DEUS showed that our variable-only translation algorithm is working successfully for classifying the UM variables. The next step will be mapping the variables in the Adaptation category based on the adaptation rules across several UMs. Future work will look into which adaptation rules each variable is used in, and if another variable and/or adaptation rule is updated by a variable’s value in order to clarify the interdependencies in the systems' UMs. Thus we have to address the following issues:

How do the variables get used within the adaptation rules? Is it possible to convert parameters that define attributes of one model so that they define the adaptation parameters of another model (and vice versa)? It will also be necessary to access the part of the system in which the adaptation rules are applied in order to establish how the conversion process should take place. Here we might need some human interaction. If this is the case, the framework should ask sensible questions that the end-user can answer and with that answer continue with the mapping if at all possible.

This framework introduces a possibility to atomize and recombine the interchange rules according to the rules of each system. This is quite an innovative technique for AH and interoperability as it is based on the use of algorithms for interdependency combinations.

References

1. Brown, E., Fisher, T., Brailsford, T.: Real users, real results: examining the limitations of learning styles within AEH. Proceedings of the HT'07, pp.57-66, Manchester, UK (2007).
2. Brusilovsky, P.: Adaptive Educational Hypermedia. Proceedings of 10th International PEG Conference, pp 8-12, Tampere, Finland (2001).
3. Celik, I., Stewart, C., Cristea, A., Ashman, H.: Interoperability as an Aid to Authoring: Accessing User Models in Multiple AEH Systems. Proceedings of the A3EH Workshop in, AH'06, Dublin, Ireland (2006).
4. Conlan, O., Wade, V., Bruen, C., Gargan. M.: Multi-model, metadata driven approach to adaptive hypermedia services for personalized e-learning. In Bra et al., p 100-111 (2002).
5. Cristea, A., De Mooij, A.: LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. Proceedings of the 14th International WWW Conference, Budapest, Hungary (2003).
6. Cristea, A., Smits, D., De Bra P.: Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia. Proceedings of the A3EH Workshop in the AIED'05, Amsterdam, the Netherlands (2005).
7. Dolog, P., Schaefer, M.: A Framework for Browsing, Manipulating and Maintaining Interoperable Learner Profiles. Proceedings of UM'05, Edinburgh, UK (2005).
8. Heckmann, D., Schwartz, T., Brandherm, B., Kröner, A.: Decentralized User Modeling with UserML and GUMO. Proceedings of DASUM in UM'05, p 61-65, Edinburgh, UK (2005).
9. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: GUMO-The General User Model Ontology. Proceedings of UM'05, Edinburgh, UK (2005).
10. IMS LIP, http://www.imsglobal.org/profiles/index.html, last visited March 2008.
11. MOT UM online version, http://wwwis.win.tue.nl/MOT05/ume/, last visited July 2006.
12. Power, G., Davis, H. C., Cristea, A. I., Stewart, C., Ashman, H.: Goal Oriented Personalisation with SCORM. Proceedings of the Fifth IEEE ICALT, p.467-471 (2005).
13. Rahm, E., Bernstein, P. A.: A survey of approaches to automatic schema matching, The International Journal on Very Large Data Bases, Vol 10, Issue 4, p 334 - 350, (2001).
14. Stewart, C., Celik, I., Cristea, A., Ashman, H.: Interoperability between AEH User Models. Proceedings of the APS Workshop in the HT'06 Conference, Odense, Denmark (2006)
15. Stewart, C., Cristea, A., Brailsford, T., Ashman, H.: Authoring once, Delivering many: Creating reusable Adaptive Courseware. Proceedings of the WBE'05, Switzerland (2005).
16. Zakaria, M. R., Brailsford, T. J.: User Modelling and Adaptive Educational Hypermedia Frameworks for Education. New Review of Hypermedia and Multimedia, 8, p.83-97 (2002).

Personalised Pervasive Modelling with a Personalised Context Ontology*

William Niu and Judy Kay

School of Information Technologies
The University of Sydney, Australia
{niu, judy}@it.usyd.edu.au

Abstract. Ontologies have been shown to effectively model domain knowledge and to automate reasoning about contextualised information. To exploit this potential in modelling a highly dynamic domain, such as pervasive computing, it is also desirable to integrate the ontological reasoning effectively with the other reasoning within the context modelling. In this paper, we describe a conceptual model that provides the means for both capturing of fast-changing contexts and reasoning about different people under various contexts. We call it a *Personalised Context Ontology* (PECO) and this model consists of four main layers: a top-level ontology, a middle ontology, an application ontology and the proposition accretion and resolution layer.
Keywords: personal ontology, ontological reasoning, pervasive personalisation

1 Introduction

Ontologies are normally used as static knowledge bases, for example, to more systematically associate tangible artefacts with abstract concepts [1] and to establish common vocabularies for knowledge sharing [2, 3]. To organise information on the magnitude of the World Wide Web, it is critical to somehow automate the information processing. To tackle this so-called *information bottleneck* problem, researchers started to semi-automate the process to construct, update and populate an ontology, which is called *ontology learning and population* [4].

Inspired by ontology learning and population, we propose to apply this technique with readily available document sources to model the highly dynamic pervasive computing environment [5]. To better capture the dynamic interaction of a pervasive computing environment, we use a dynamic context ontology with the *accretion and resolution* (A/R) approach, which provides flexibility and effectiveness in modelling such an environment [6]. This enables us to create a context ontology that can be continuously updated and provides personalised and contextualised information in a pervasive computing environment. We called it a *Personalised Context Ontology*, or PECO.

* The Smart Internet Technology CRC partially funded this project.

2 Related Work

One of the cornerstones of interoperability is the definition of a vocabulary or ontology that can be re-used across applications. In the context of pervasive computing (PerCom), there has been considerable diversity in the approaches explored for defining and exploiting such an ontology. The related work of addressing problems of different sets of context ontologies has also explored, for example in GAIA [2]. The role of ontologically supported reasoning in PerCom and context modelling has also been examined, for example in the Semantic Space [7] and by Chen, Finin and Joshi [3]. Departing from that body of work, we have taken an approach that is driven by the goal of scrutability; certainly our ontology should serve the various reasoning needs of PerCom, such as reasoning about inconsistent evidence about location [8] and to provide personalised information about a place [9], but we also want to design the ontology so that it can support explanations of the reasoning. We have created and evaluated an application that does this [9]. In this paper, we described the conceptual framework underlying the design of the ontology, including the lower, more application-specific levels.

In particular, we introduce a new form of "personal ontology", which takes the view that different people's understanding of a pervasive environment are different and that an ontology can capture this. Previous work has explored various issues about personal ontologies, including but not limited to personal ontology comparison [10], personal profiles for Web navigation [11], and personal ontology elicitation and visualisation [12]. Our work tackles the challenge of creating low cost, pragmatic personal ontologies by a semi-automated process, based on mining resources that capture ontological information.

3 A Conceptual Model of PECO

Figure 1 illustrates a layered model for PECO. The following is the four main layers: a *top-level ontology* that constitutes the most general and formal ontology; a *middle ontology* that is application-independent for a small set of domains; an *application ontology* that specifies vocabulary relevant to a particular application; and the most specific and light-weight layer of *proposition accretion and resolution*.

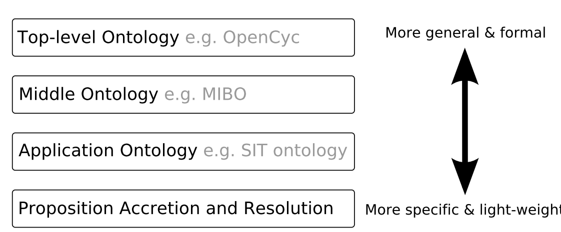


Fig. 1. A layered model for PECO

3.1 Top-level Ontology

A top-level ontology specifies concepts that are generally true across all domains. This is the top-most layer of PECO. Examples of top-level ontologies include the

Cyc [13], Wordnet [14], and SUMO [15]. In our work, we have used OpenCyc, a subset of Cyc, covering some of the most general notions and relationships.

Building an ontology from or referring to a manually-built generic ontology has several advantages, including re-usability, interoperability, automatic reasoning, and wide acceptance.

3.2 Middle Ontology

In practice, a generic ontology is often too general and too computational expensive for modelling and reasoning about a relatively small set of concepts. As a result, a more specialised and domain-specific ontology is often more suitable, especially for real-time reasoning. This ontology can consist a relatively small set of concepts and relationships that can be crafted from the top-level ontology. We call this a *middle ontology*. There are several proposals for such ontologies across different domains: for example, SOUPA [3] for context modelling in PerCom environments; GUMO for user modelling for the Semantic Web [16]; and MIBO for indoor location modelling [5]. In PerCom, this can be the ontology that defines the key concepts about any building, including concepts such as *floor*, *wing*, *level* and *room* and relationships between such as *floor is-part-of building* or *building is-a fixed structure*.

3.3 Application Ontology

An *application ontology* contains the application-specific vocabulary, which is more specialised than both the top-level and middle ontologies and often cannot be re-used [17]. At this layer of PECO, the aim is to construct a base ontology by carefully analysing the middle ontology and an authoritative document source, such as a building plan, which would provide a foundation for populating application-specific concepts and relationships. It is important that this process can be semi-automated. This makes it easier to ensure that it is more complete and systematically created so that changes in the PerCom environment can be readily integrated into the representation over time. One important way to do this is to exploit existing document sources.

An example of an ontology of this model layer is the *SIT ontology* described in [5]. The SIT ontology extends from the MIBO and is generated from the building maps represented in the Scalable Vector Graphics (SVG) format. The ontology exists in both a popular ontology language: OWL-Lite, as well as in a less conventional, but more suitable format for ontological reasoning in the PerCom environment, PersonisAM models [18].

Note that while the ontology created at this application level is not re-usable across different contexts, the tools that create it may be. In our case, the ontology that we create captures relevant relationships about one specific building. However, the process and code used to create it from the building maps could be used for any other building for which there is a similar representation in Scalable Vector Graphics (SVG) format. Similarly, a similar program could be created to operate in conjunction with the building model representation used in the various Computer-Aided Design packages that architects use. This means that while the ontology is highly specific, the process for creating it may be re-usable.

3.4 Proposition Accretion and Resolution: Ontology Population

Before describing this layer of PECO model, we briefly describe the *accretion and resolution* (A/R) approach to paint a background. A/R has been designed for simplicity, flexibility and, critically, for user control and scrutability. It has two fundamental steps: *accreting* evidence and *resolving* a value at the time, and in the context, that a value is needed. This approach forms the the bottom layer of the model and can serve as a bedrock for reasoning about personalised and contextualised information in a PerCom environment.

In PECO the accretion step happens when the propositions found from various sources are populated onto the base ontology. As each proposition is extracted, it is simply stored as evidence with a timestamp and a source in the relevant concept without further interpretation. The proposition may, for example, be from a sensor detecting a mobile device (e.g. Bob's mobile phone is at Room 123), a staff directory (e.g. Bob has an email address bob@example.com) or an assertion by a user (e.g. Bob is at Room 123).

Later, when a value is required for a concept in a proposition (e.g. where is Bob?), the stored evidence is examined and *resolved* to a value. The resolution process can be personalised depending on the context of the program or person requiring the value. This is done by running a customised *resolver* function. For example, for someone who usually carries a mobile phone, evidence for the mobile phone's location is valuable for determining that person's location; but for a person who often forgets to carry the mobile phone, the evidence would not be as reliable, and the resolver function may account for that fact. Taking another example, Bob may only want to release his location information to the general public at a relatively coarse grain-size, say, at the level of the building. For this class of person, he may also choose to exclude evidence generated by their mobile phone. This can be done by using a different resolver function when releasing Bob's information to the general public.

4 Discussions

The use of the PECO framework is demonstrated in [9]. It demonstrates the potential of such an ontology in delivering personalised information, as well as in providing scrutability about personalisation, which is a critical step to build an acceptable PerCom system. The design of PECO is intended to ensure that our applications which make use of it could be readily ported to a new building. If, for example, our School were to move to a new building, we should be able to directly re-use the MIBO. In addition, if we had access to similar SVG maps, we should be easily able to automatically create the application ontology for that new building. Some of the documents and resources we use to populate the A/R level would be directly applicable in the new building. In addition, once the other documents were updated to provide their information for the new building, as would need to be done, we could rebuild this layer from those documents.

We have outlined the four-level conceptualisation of PECO for reasoning in PerCom environments. We have described the motivation for defining each of these layers and discussed the re-usability at each level. Our work is distinguished by our particular concern for scrutability and support for control over privacy, a particularly important issue for PerCom.

References

1. Hatala, M., Wakkary, R., Kalantari, L.: Rules and ontologies in support of real-time ubiquitous application. J. of Web Semantics **3** (2005) 5–22
2. Ranganathan, A., McGrath, R.E., Campbell, R.H., Mickunas, M.D.: Use of ontologies in a pervasive computing environment. Knowl. Eng. Rev. **18** (2003) 209–220
3. Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard ontology for ubiquitous and pervasive applications. In: MOBIQUITOUS '04, (2004) 258–267
4. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems **16** (2001) 72–79 Special Issue on Semantic Web.
5. Kay, J., Niu, W., Carmichael, D.J.: ONCOR: Ontology- and evidence-based context reasoner. In: IUI '07, (2007) 290–293
6. Carmichael, D.J., Kay, J., Kummerfeld, B.: Consistent modelling of users, devices and sensors in a ubiquitous computing environment. User Modeling and User-Adapted Interaction **15** (2005) 197–234
7. Wang, X., Dong, J.S., Chin, C., Hettiarachchi, S., Zhang, D.: Semantic Space: An infrastructure for smart spaces. IEEE Perv. Comp. **3** (2004) 32–39
8. Niu, W., Kay, J.: Location conflict resolution with an ontology. In: PERVASIVE '08, (2008) 162–179
9. Niu, W., Kay, J.: Pervasive personalisation of location information: Personalised context ontology. In: AH '08, (2008)
10. Dieng, R., Hug, S.: Comparison of personal ontologies represented through conceptual graphs. In: ECAI '98, (1998) 341–345
11. Gauch, S., Chaffee, J., Pretschner, A.: Ontology-based personalized search and browsing. Web Intelligence and Agent System **1** (2003) 219–234
12. Katifori, A., Vassilakis, C., Daradimos, I., Lepouras, G., Ioannidis, Y., Dix, A., Poggi, A., Catarci, T.: Personal ontology creation and visualization for a personal interaction management system. In: Workshop on PIM, in CHI 2008, (2008)
13. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM **38** (1995) 33–38
14. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to WordNet: An on-line lexical database. Int. J. of Lexicography **3** (1990) 235–244
15. Niles, I., Pease, A.: Towards a standard upper ontology. In: FOIS '01, (2001) 2–9
16. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: GUMO - the general user model ontology. In: UM '05, (2005) 428–432
17. Guarino, N.: Formal ontologies and information systems. In: FOIS '01, (1998) 3–15
18. Assad, M., Carmichael, D., Kay, J., Kummerfeld, B.: PersonisAD: Distributed, active, scrutable model framework for context-aware services. In: PERVASIVE '07, (2007) 55–72

Author Index