Proceedings of the International Workshop on

Architectures and Building Blocks of Web-Based User-Adaptive Systems (WABBWUAS)

co-located with
International Conference on User Modeling, Adaptation and Personalization (UMAP) 2010, Hawaii, USA

Editors: Fabian Abel[1], Eelco Herder[1], Geert-Jan Houben[2], Mykola Pechenizkiy[3], Michael Yudelson[4]

[1] IVS – Semantic Web Group & L3S Research Center, Leibniz University Hannover, Germany
{abel,herder}@L3S.de
[2] Web Information Systems, TU Delft, The Netherlands
{g.j.p.m.houben}@tudelft.nl
[3] Information Systems Group, Eindhoven University of Technology, The Netherlands
{m.pechenizkiy}@tue.nl
[4] School of Information Sciences, University of Pittsburgh, USA {mvy3}@pitt.edu

Table of Contents

Preface

User-adaptive systems have evolved from small-scale stand-alone applications to interactive Web-based applications that are often deployed on a larger scale. Consequently, the need has arisen to move from prototypical systems to scalable, deployable solutions. At the same time, a shift can be seen from rule-based, mentalistic user modeling approaches to 'Web 2.0' approaches that involve machine learning, data mining, and collaborative techniques.

Past research provided a large body of methods for adaptation/personalization, and techniques for user modeling, usage mining, and collaborative filtering. Conceptual frameworks splitting the adaptation process into various layers provide guidance for implementing user-adaptive systems. Based on these building blocks, various groups have created their own frameworks, among others AHA!, APELS, and Personal Reader. Framework design provides an opportunity to reuse components or even whole layers of the adaptation process. Reuse of components such as user behavior observation and logging tools, user model storage promotes faster development, better feature selection, and more robust systems.

Although, system fragmentation enables component reuse and speeds up the development of the new systems, there are several issues. First, decomposition of a monolithic system should result in a good abstraction of the data and process model to provide a convenient basis for reuse. Second, the data traffic between the separated system components may intensify. As the number of system users increases issues related to scalability might arise. This is especially true for user-adaptive and cognitive systems where the modeling and personalization components are traditionally computationally and data intensive.

Existing work on the Web-based user-adaptive and cognitive systems, including work on frameworks, shown that there exists a strong overlap between conceptual models of the decomposed adaptation process and the practical implications of its design. In this situation, a logical step is to compare already working systems with emerging approaches and models. In this workshop we seek to identify current practices and experiences with concrete implementations of user-adaptive and cognitive systems or specific components - varying from experimental, small-scale prototypes to systems that are deployed on a larger scale.

The topics of this workshop include but are not limited to:

- user behavior observation and user data collection: embedded into the adaptive system or available as standalone components or add-ons,
- user data management: data storage platforms and formats, the use of open standards, querying techniques or APIs, interoperability issues,
- reusing reasoning and adaptation techniques,
- scalability and performance issues of user modeling and adaptation,
- generalizable techniques for adaptation, personalization and recommendation,
- translations of conceptual designs into concrete implementation,
- deployment issues and lessons learned (case studies and evaluation).

In summary, research papers presented at this workshop focus on *Architectures and Building Blocks of Web-Based User-Adaptive Systems*.

We would like to thank the members of the Program Committee of WABBWUAS 2010 for their support and reviews. Further, we are grateful to all authors who submitted articles to WABBWUAS and contributed with their works to the WABBWUAS workshop.

Fabian Abel
Eelco Herder
Geert-Jan Houben
Mykola Pechenizkiy
Michael Yudelson
WABBWUAS Organizing Committee, June 2010

Organization

WABBWUAS 2010 is co-located with the International Conference on User Modeling, Adaptation and Personalization (UMAP) 2010, Hawaii, USA.

Organizing Committee

Fabian Abel
Eelco Herder
Geert-Jan Houben
Mykola Pechenizkiy
Michael Yudelson

Program Committee

Paul De Bra (Technical University of Eindhoven, the Netherlands)
Peter Brusilovsky, Peter (University of Pittsburgh, USA)
Evgeny Knutov (Technical University of Eindhoven, the Netherlands)
Kevin Koidl (Trinity College Dublin, Ireland)
Daniel Krause (University of Hannover, Germany)
Ian O'Keeffe (Trinity College Dublin, Ireland)

Student modeling services for hybrid web applications

Iván Cruces[1], Mónica Trella[1], Ricardo Conejo[1], Jaime Gálvez[1]

[1] Dpto. Lenguajes y Ciencias de la Computación.
Universidad de Málaga 29071 - Málaga (Spain)
{icruces, trella, conejo, jgalvez}@lcc.uma.es

Abstract. This paper introduces a set of resources that provide web learning environments with student modeling services. SAMUEL is a user modeling server for registering, updating and maintaining student knowledge data from different sources that use their own ontologies. In order to make inferences about student knowledge, it becomes necessary to establish equivalences between concepts of different domains. For this purpose we have developed SONIA, a tool to edit and integrate models which allows semi-automatic ontology mapping. The set of resources is completed with INGRID, an independent generic open learner model for interacting with external student models.

Keywords: Web Intelligent Learning Environment, Student Modeling Server, Open Learner Model

1 Introduction

In the last decade, several adaptive and intelligent web-based learning environments (WILE) have been developed. Most of these systems are the result of research efforts focused on a particular pedagogical task, learning domain or teaching strategy, like for example ELM-ART [1] or AlgeBrain Tutor [2] which deal with LISP and Algebra domains respectively. Although less common, there are also educational web-based tools for generic domains such as SIETTE [3] or DCG [4]. Given the availability of all these tools, educators and course designers may be interested in integrating some of them in their own courses. Since these and other systems are high-quality software based on solid theoretical foundations, they may be of great value for the development of other educational systems. However, modularity is limited in most cases, which makes reusability almost impossible unless the system is used as a whole.

When thinking about developing tools that allow reusability of existing components, desirable features for such tools are: domain-independence (tools can be used for any subject domain); extensibility (in the sense that any component can be integrated), and component interoperability (components can communicate and interoperate despite differences of implementation language, execution environment

[1] This work has been partially supported by the ARTEMISA project number TIN 07-67515 of the MINISTERIO DE CIENCIA E INNOVACIÓN, Spain.

or model abstraction). Interoperability can be approached from two different angles: as a distributed software problem, and as a semantic conceptual issue.

Nowadays, web-based learning environments are evolving by adapting their architectures to new Web 2.0 technologies. The new approaches rely on the development of distributed architectures based on the integration and reuse of learning activities. Consequently WILEs become hybrid web applications (mashup).

Furthermore, the web teaching/learning model evolves towards an auto-learning scenario in which students complete their training using resources located all over the web. In any case, we deal with students who are using different systems. If these systems are intelligent (i. e. manage their own student model), each one of them stores part of the information about the student's knowledge, that is, the student model is distributed over the network. In this context, it would be useful that a system "asks someone for references" when it has to work with a new student, in order to provide adapted instruction.

This problem can be approached by using user modeling servers (UMS) [5;6;7]. According to Kobsa [8], *the purpose of user modelling servers is to separate user modelling functionality from user-adaptive application systems. They are not a part of an application system but rather independent from it*. In this way, an UMS is part of a distributed learning environment which provides teachers and users with educational services. ADAPT^2 [9] is an example of a framework for distributed education that integrates an UMS as part of its architecture. It stores students' activity and infers student's knowledge. When a student interacts with systems that have their own UMS, ADAPT^2 facilitates user models integration by means of Ontology Servers. It allows the knowledge of the same student to be modeled by multiple systems along different ontologies and stored on different ontology servers. However, once several adaptive systems decide to collaborate in sharing and exchanging student models, they have to select one specific ontology ;then the server used for user model exchange will be the one that hosts the selected ontology.

Our research group have been working during the last years in MEDEA [10], a distributed framework for the integration of web-based educational systems. MEDEA provides authors with the core components for an intelligent learning system: domain, student and instructional modules. MEDEA provides students with the necessary guidance during the learning process. It decides at each moment the most adequate task to be performed by the student. These tasks are performed by external learning resources (LR). Early version of MEDEA uses a central ontology and allows to manually map the domain model of LR into it. So, the user information received from LR could be translated into the concepts of MEDEA's ontology in order to update the user model. One of the main weaknesses of this version is that LRs cannot share data among them. They can only update the central user model but none other system can take advantage of it. Second, semantic integration of external adaptive systems depends on a manual mapping which is a costly task that needs the intervention of teachers and domain experts. In order to address these issues, MEDEA has evolved toward a more decentralized architecture. A set of resources that provides web learning environments with students modeling services has been developed and is presented in this paper.

SAMUEL (Spanish acronym for User Modeling Accumulative Server for E-Learning), is a User Model which allows storing student knowledge evidences obtained from different learning resources. SAMUEL is an independent component that can be requested by any web LR that needs to obtain data about a certain student.

SAMUEL stores heterogeneous information from different systems that deal with different domain ontologies. In order to perform evidence integration it is necessary that systems agree on the semantic of the domain terms, so they can exchange data for the equivalent domain concepts to update their own models.

Some attempts can be found in the literature for user model integration. In [11] authors propose a conversational model for reaching an agreement over not shared concepts. An ontology-based approach is used in [12] to identify similar concepts in the ontologies of related domains and align the domain models of two adaptive educational systems. In this paper we present an initial proposal based on ontology mapping techniques [13]. For this purpose SONIA (Spanish acronym for Intelligent Ontology Server for E-Learning) has been developed. This tool allows domain model edition and semi-automatic ontology mapping. SONIA stores the domain ontologies of external LR. When a student's mark in a domain concept is requested from Samuel, SONIA provides a list of equivalent domain terms. Then evidence integration heuristics are applied to all the concepts included in the list.

The set of student modeling services is completed with INGRID (Spanish acronym for Domain Independent Graphic Interface). This system allows students to consult (via Web) user models of resources with which they work and interact.

In the next section a description of different usage scenarios of MEDEA is discussed. In the rest of the paper, each tool is described in detail: SAMUEL (user modeling server) in section 3; SONIA (ontology mapping tool) in section 4, and INGRID (Open Learner Model) in section 5. Finally some conclusions of this work are presented.

2 MEDEA scenarios

In this section we describe new MEDEA architecture through different usage scenarios.

MEDEA provides all the components needed to create and execute an Intelligent Learning Environment: support for Domain Model definition, User Model (data storage and diagnosis processes), Instructional Planner and User Interface. All of them can be used together to act as a WILE while some of them can be used independently by any Educational Software.

The figure 1 shows the scenario 1, where all MEDEA components are used (MEDEA planner isn´t introduced in this paper). Teachers create the domain model and decide which learning resources are adequate to each concept (dotted line). Then a student interacts with the WILE through a web browser as described below.

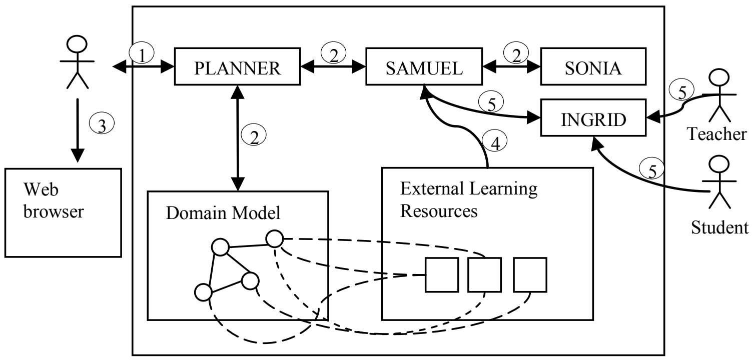


Fig. 1. MEDEA architecture - Scenario 1

Scenario 1

1. Student asks MEDEA planner for advice.
2. MEDEA planner consults domain model and student model (SAMUEL). SAMUEL uses SONIA to obtain all the data related to MEDEA domain model concepts. Each domain topic is related to one or more external LR that can be used to learn it. MEDEA planner suggests next topic to be learnt and some tasks (LR) to be done by the student..
3. Student works with a LR.
4. LR updates student model invoking SAMUEL web services.
5. Teachers and/or students can consult the student model using INGRID (Graphic Interface).

Besides being used together, each resource can constitute an independent component which may be used in combination with an LR. MEDEA has therefore evolved into a mashup. The figure 2 shows another possible scenario where some MEDEA components are used by external resources.

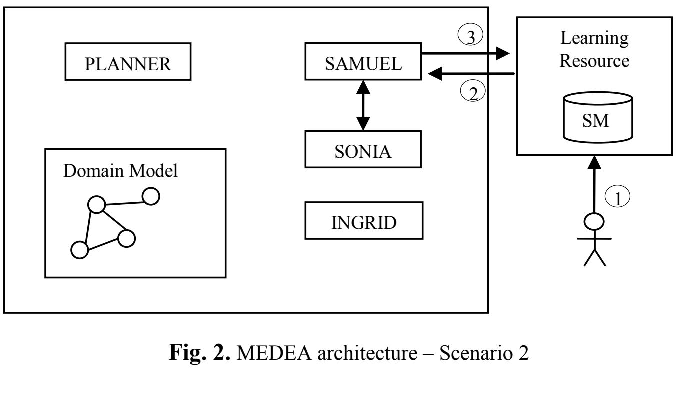


Fig. 2. MEDEA architecture – Scenario 2

Scenario 2

1. Students works independently with a LR .
2. The LR asks SAMUEL for information about this student's knowledge of a concrete topic. SAMUEL, using SONIA, compiles all the available data and sends it to the LR.
3. When the student finishes, the LR updates its own student model and can update SAMUEL too.

The figure 3 describes how INGRID can be used by a LR that has no way to show graphically its student data.

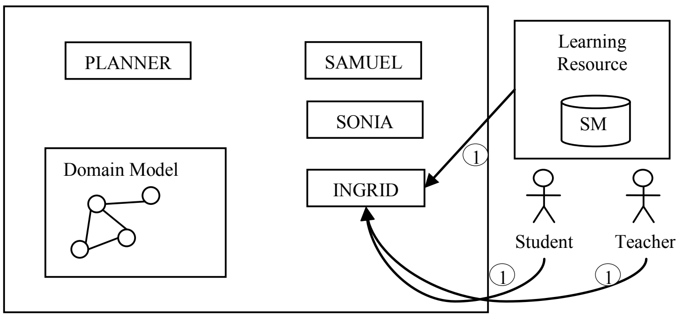


Fig. 3. MEDEA architecture - Scenario 3

Scenario 3

1. An external LR uses INGRID as a display for its model.

3 User model server for e-learning

SAMUEL is a user modeling server that allows storing of student knowledge evidences about different domain concepts obtained from different learning resources (Figure 4). Each LR can manage its own domain model, so each topic (domain concept) mark is stored independently together with the student ID, the domain and the source LR Therefore it is necessary, in order to make inferences about student knowledge, to establish equivalences between concepts of the same domain that belong to a different ontology. For this purpose we have developed an Intelligent Ontology Server for E-Learning (see next section).

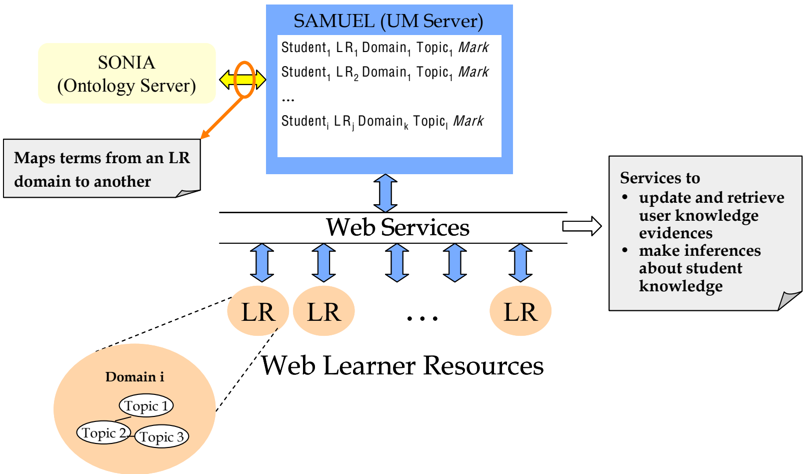


Fig. 4. SAMUEL architecture

For example, if we consider two WILEs which have a domain related to economics, we could map the terms as in Table 1.

Table 1. Economic domain ontology.

WILE 1
Topic 1: Economic concepts Topic 2: Kinds of markets
WILE 2
Topic 1: Introduction Topic 2: Financial system Topic 3: Markets …
Maps terms
(WILE 1) Topic 1: Economic concepts <-> (WILE 2) Topic 1: Introduction (WILE 1) Topic 2: Kinds of markets <-> (WILE 2) Topic 3: Markets …

If a student performs a task with WILE 1 about *Kinds of markets* topic and a task with WILE 2 about *Markets* topic, both, WILE 1 and WILE 2, will register the evidences in SAMUEL. So, if WILE 1 requests the mark of student in *Kinds of markets* topic, SAMUEL will infer it with *Kinds of markets* evidences and *Markets* evidences, through the mapping terms.

At present SAMUEL offers a set of web services implemented with JAX-WS API, which allows web learning resources to register evidences and to obtain data about other user activities. Besides a concept mark, each record contains information about the learning activity that provides the information. The services have three main parameters: the set of evidences used to estimate the student's knowledge, the evidence sources and the method used to obtain the estimation.

Table 2. Parameter values of SAMUEL services.

Nº evidences	Source	Estimation
Last evidence	Specific source	Average mark
Last n evidences	A set of sources	Weighted mark
Evidences in a date range		

According to the values showed in Table 2, SAMUEL can be asked to return the *last n* evidences concerning a concept or those collected in a period of time. In both cases, clients can retrieve average and weighted marks for any concept. These data can be referred to one or more learning resources. The heuristics used to calculate the weighted marks are represented by equation **1**.

$$mark(c) = \frac{\sum_{i=1}^{n} w_i m_i}{\sum_{i=1}^{n} w_i} \quad \textbf{(1)}$$

In *Evidences in a date range* case the *n* value in equation **1** represents the total number of days in the date range and the m_i value is the average mark of day i. In *Last n evidences* case the *n* value represents the n parameter and the m_i value is the mark of the *i-th* last evidence registered. The w_i values are the mark weights. One example could be $w_i=1/i$, where more weight is given to the most recent evidences since they represent more accurately the student's current knowledge.

4 SONIA

As it has already been discussed, in order to make inferences about student knowledge using data collected from different learning resources, it becomes necessary to establish equivalences between concepts of different domains. We have approached this problem with the development of a web ontology server called SONIA. It is an AJAX application that allows editing concept semantic networks (first level ontologies) and establishing relations among them semi-automatically (Figure 6).

Stemming algorithms (for English and Spanish languages) are applied to ontology terms and then a set of *string similarity metrics* (Hamming distance, N-gram, Levenshtein distance and Maximum common substring) are used to calculate the probability of two terms from different ontologies referring the same concept. In addition to these metrics, a synonyms dictionary has been added.

The Figure 5 shows an example where terms of two ontologies are compared. Each arrow's color represents a different metric. The number above the arrow indicates the similarity degree between the terms. These correspondences can be updated (inserted and/or deleted) manually.

These mappings are stored and, when SAMUEL is requested information about a concept (Figure 4), it searches for all the terms related in order to collect data which refer to the same concept and have been provided by different WILEs.

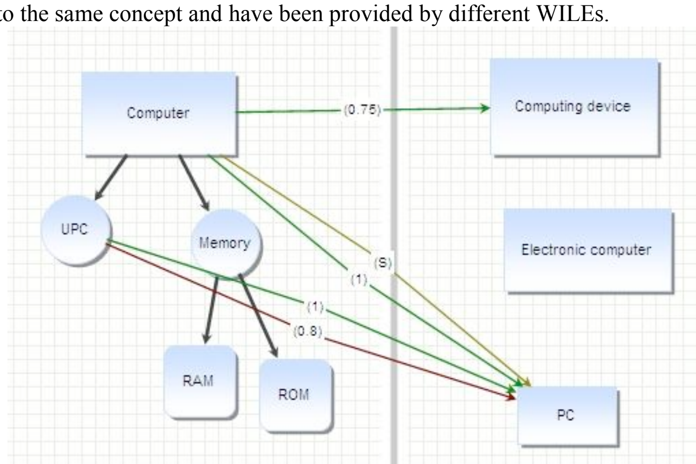


Fig. 5. Ontology mapping in SONIA.

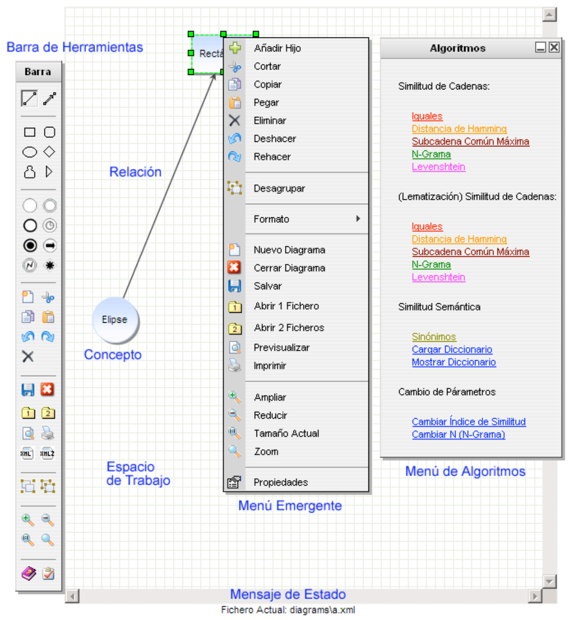


Fig. 6. SONIA interface.

5 Domain independent graphic interface for student models

Open learner models (OLM) are accessible and open models that extend the traditional WILE models to turn them into a visible and interactive part of the system [14]. OLM allows a student to inspect his model and interact (edit or negotiate) with it. This kind of system stimulates the student's analytical thinking and helps him/her to plan and monitor his/her learning [15]. OLMs are not just valuable for students but also for teachers. In fact, a graphical representation of the model can help teachers to carry out a course formative evaluation and to determine the students' learning problems. Moreover, the model accuracy can be improved if the system allows students and teachers to collaborate in the modeling process. Broadly speaking, the student model has evolved from being a knowledge source for learning resources (closed system) to become an important learning resource for the student (open systems).

There are two types of OLM: those integrated in a tutor system and the independent ones (IOLM), whose purpose is to help students to identify and to solve learning problems by themselves, without tutor system help, that is, to encourage metacognitive skills.

Still in the field of resource integration, we have developed INGRID [16], a web IOLM which allows students to consult the user models of the different learning resources with which they work and interact. INGRID provides two views of student score, both based on the relationship topic / sub-topic from the domain concepts. The first is a hierarchical structure (Figure 7) representing the issues by a graph, and the second is a table of topics and marks. The hierarchical view of the graph represents the nodes with a color code indicating the student's level of knowledge in this concept according to a particular source (SIETTE system, for example). The table of topics shows bars that represent the marks on a scale from one to ten. Moreover, each topic can have several actions associated to it to edit the user model (e.g. SIETTE tests).

The strengths of the system are that it is generic and it can represent data from any WILE. For this purpose, INGRID has a JSP which receives as input (xml format) a list of concepts and the marks obtained by the students, as well as the semantic network of the domain model (concepts and relationships) and is capable of representing it. So far it has been successfully tested with SIETTE and user model server SAMUEL. Both tools use it as a plug-in to graph the data from their students.

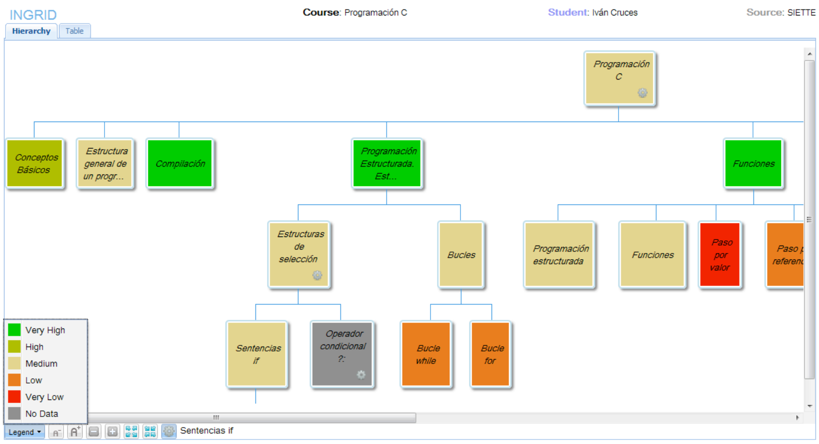


Fig. 7. Hierarchical view in INGRID.

6 Conclusions

The general objective of our research work is the development of different domain independent and interoperable components that can be used in the construction of a web based intelligent learning management systems.

The user interaction with different learning resources can provide valuable information that helps other systems to provide the student with a more accurate instruction. In this sense we work in the integration of heterogeneous sources of user information. In order to achieve this, our group is working on the development of independent and generic tools that provide student modeling services. A functional approach has been presented in this paper. It includes a set of tools that can be used in any learning environment to share user data.

In this first stage of the work, all the tools needed to provide user modeling services have been implemented and all of them are working successfully in a real environment. As part of a formative evaluation process, these tools have been integrated with the test system SIETTE [17]. At present, we are gathering user data from SIETTE, and we are planning to use new learning resources as data sources in the near future. The results obtained from this evaluation have opened up possibilities for future research.

First, we are aware of the importance of the semantic issue in WILE interoperability. It is a bottleneck in intelligent learning resources integration. A first approach to this problem is SONIA. So far, we have applied ontology mapping techniques based on lexical components. The next step is to use techniques based on ontology structure as graph matching and studying relations semantic. In the next stage of this work we plan to conduct research into the application of machine learning or statistical techniques for ontology mapping.

Furthermore, we have implemented a set of services to accumulate user evidences. We plan to add new services which apply formal diagnosis methods to make inferences about student.

Finally, we think that teachers and students are an important source of evidences, therefore we will not only allow INGRID to consult the server data but also to update them taking into account teachers and students' contributions.

References

1. Weber, G. and Brusilovsky, P.: ELM-ART: An adaptive versatile system for Web-based instruction. Special Issue on Adaptive and Intelligent Web-based Educational Systems. International Journal of Artificial Intelligence in Education, vol. 12, pp. 351--384 (2001)
2. Alpert, S. R., Singley, M. K., and Fairweather, P. G.: Deploying Intelligent Tutors on the Web: An Architecture and a Example. International Journal of Artificial Intelligence in Education , vol. 10, pp. 183--197 (1999)
3. Conejo, R., Guzmán, E., Millán, E., Pérez-de-la-Cruz, J. L., and Trella, M.: SIETTE: A Web-Based Tool for Adaptive Testing. International Journal of Artificial Intelligence in Education, vol. 14, pp. 29--62 (2004)
4. Vassileva, J. and Deters, R.: Dynamic courseware generation on the WWW. British Journal of Educational Technology, vol. 29, pp. 5--14 (1998)
5. Kay, J., Kummerfeld, B., and Lauder, P.: Personis: A server for user modeling. Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), 201--212. Málaga, Spain (2002)
6. Brusilovsky, P., Sosnovsky, S., and Shcherbinina, O.: User Modeling in a Distributed E-Learning Architecture. In: L. Ardissono, P. Brna and A. Mitrovic (eds.) Proceedings of 10th International User Modeling Conference (Edinburgh, UK, July 24-29, 2005). Lecture Notes in Artificial Intelligence, vol. 3538, pp. 387--391. Springer-Verlag, Heidelberg (2005)
7. Kobsa, A., & Fink, J. (2006). An LDAP-based User Modeling Server and its Evaluation. User Modeling and User-Adapted Interaction 16(2), 129-169.

8. Kobsa, A.: Generic User Modeling Systems, in The Adaptive Web. LNCS, vol. 4321, pp. 136--154. Springer-Verlag, Heidelberg (2007).
9. Brusilovsky, P., Sosnovsky, S., and Yudelson, M. (2005) Ontology-based framework for user model interoperability in distributed learning environments. In: G. Richards (ed.) Proceedings of World Conference on E-Learning, E-Learn 2005, Vancouver, Canada, October 24-28, 2005, AACE, pp. 2851-2855.
10. Trella, M., Conejo, R., Guzmán, E. y Bueno, D. (2003): An Educational Component Based Framework for Web ITS development. Web Engineering. International Conference, ICWE 2003. LNCS, vol. 2722, pp. 134--143. Springer-Verlag, Heidelberg
11. Cena, F., Furnari, R. (2008): A soa-based framework to support user model interoperability. In: Nejdl, W., Kay, J., Pu, P., Herder, E. (eds.) AH 2008. LNCS, vol. 5149, pp. 284–287. Springer, Heidelberg
12. Sosnovsky, S., Dolog, P., Henze, N., Brusilovsky, P., & Nejdl, W. (2007). Translation of overlay models of student knowledge for relative domains based on domain ontology mapping. In: R. Luckin, K. R. Koedinger, & J. Greer (eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED'2007), Marina Del Ray, CA, USA, July 9-13, 2007*, 289-296.
13. Y. Kalfoglou, M. Schorlemmer (2003). Ontology Mapping: The State of the Art.
14. The Knowledge Engineering Review 18 (1):1-31, 2003.Bull, S., Kay, J. (2007): Student Models that Invite the Learner In: The SMILI:() Open Learner Modelling Framework. International Journal of Artificial Intelligence in Education, vol. 17, pp. 89--120
15. Mitrovic, A., Martin, B. (2002): Evaluating the Effects of Open Student Models on Learning. LNCS, vol. 2347, pp. 296--305. Springer-Verlag, Heidelberg
16. INGRID, http://marte.lcc.uma.es/ingrid/index.jsp&demo=1
17. SIETTE, http://www.siette.org
18. MEDEA, http://marte.lcc.uma.es/medea

Generic Adaptation Process

Evgeny Knutov, Paul De Bra, and Mykola Pechenizkiy

Department of Computer Science, Eindhoven University of Technology
e.knutov@tue.nl, debra@win.tue.nl, m.pechenizkiy@tue.nl
P.O. Box 513, 5600 MB, Eindhoven, the Netherlands

Abstract. Adaptive Hypermedia Systems (AHS) have long been mainly represented by domain- or application-specific systems. Few reference models exist and they provide only a brief overview of how to describe and organize the 'adaptation process' in a generic way. In this paper we consider the process aspects of AHS from the very first classical 'user modelling-adaptation' loop to a generic detailed flowchart of the adaptation in AHS. We introduce a Generic Adaptation Process and by aligning it with a layered (data-oriented) AHS architecture we show that it can serve as the process part of a new reference model for AHS.

1 Introduction and Background

Throughout the development of the Hypermedia and later Adaptive Hypermedia (AH) research field people have been trying to create 'reference' models of these categories of systems. Major reference models have been favouring a layered architecture, starting with the Dexter Hypertext Model [7], and later the Tower Model [5] (introduced as the Extensible Data Model for Hyperdocuments) and this was continued in adaptive hypermedia with the most referenced AHAM [6] model, followed by other systems/models, such as LAOS [11] (elaborating layered approach), APeLS [4], the Munich model [10], etc. However, these developments were mostly concerned with the structure and/or the data model, but not as much with the process underlying the adaptation.

In the paper we examine the issue of aligning the adaptation process, based on an extensive list of AH methods and techniques [9], with a layered structure of AHS. We show that to some extent the process influences and defines the composition and the sequence of such a layered structure in such a way that it partially arranges the order of the layers, defines couplings and determines the major transitions in the system. We show that the process driven approach gives more insight in AH development methods and the composition of the AH system.

1.1 Adaptation Process Modelling

Hereafter by *Generic Adaptation Process* we mean the interaction in AHS which starts with the goal statement, exploits features of the user and domain models in different contexts and adapts various aspects of the information and presentation to the user. Figure 1 shows this user modeling / adaptation loop as originally presented in [3].

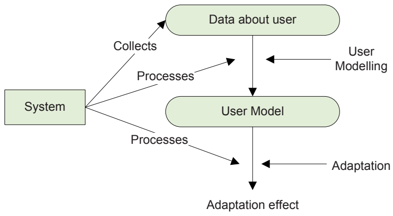
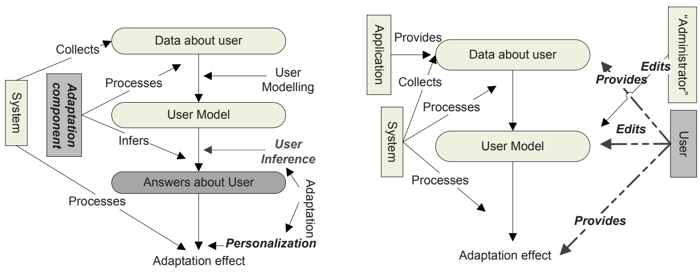


Fig. 1. Classic loop user modelling - adaptation

Considering a generic adaptive system one may think not only about defining a framework or reference (data) model but also about what the adaptation process within the system looks like, beyond what Fig. 1 shows. Fig. 2 shows some extensions of the classical loop, taking into account that selection of user information or reasoning about the user model to obtain answers about the user is an essential part of the adaptation process (Fig. 2a) and that either the user or an administrator (or both) need the ability to scrutinize the user model (Fig. 2b).



(a) User model inference - adaptation loop

(b) Collaborative user model editing (User and Administrator involved)

Fig. 2. User modelling - adaptation loops

These updated 'user-modelling - adaptation' loops give a more extensive overview of some aspects of the adaptation, however in [9] we integrated the entire classification of AH methods and techniques (see Sect. 2) with the adaptation process cycle to give a first insight into the generic adaptation process flow, see Fig. 3.

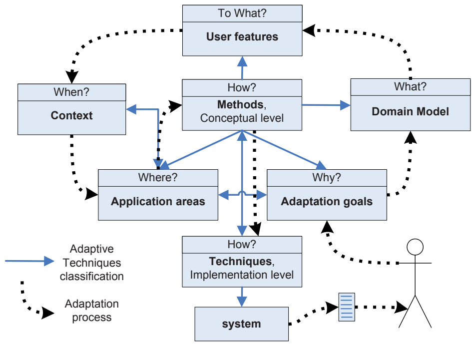


Fig. 3. Classification of AH methods and techniques; adaptation process highlights [9]

Although coupling the AH methods classification with the 'adaptation process' had a different purpose from what is shown in the classical (and later) loops of 'user modelling - adaptation' - our goal is to show the diversity of the adaptation process representation and the possibility of aligning not only the 'user-adaptation' loop into the adaptation process but the adaptation methods and techniques as well.

1.2 Goals

In this paper we describe the reference adaptation process, aligning it with the traditional 'adaptation questions' (Sect. 2) and formalizing it in a single generic manner. In particular, we:

- *provide a flowchart diagram of a generic AHS (based on the summarization from [9])*;
- *put the notion of the adaptation process in a context of a generic layered adaptation system*;
- *align the layers of AHS in a sequence chart and present the reference adaptation process.*

2 Questions of Adaptation and Adaptation Sequence

Adaptation can be defined by posing and answering six major questions:

- Why do we need adaptation? (*Why?*)
- What can we adapt? (*What?*)
- What can we adapt to? (*To What?*)
- When can we apply adaptation? (*When?*)
- Where can we apply adaptation? (*Where?*)
- How do we adapt? (*How?*)

This type of classification has been initially introduced in [3]. Here we not just revisit these questions, but address the issue of aligning them (also aligning the corresponding methods, techniques and respective modules (layers) of AHS) in a *generic adaptation process* which can serve as a process guideline and framework for defining the way AHS functions.

Fig. 3 considers the order in which the adaptation questions should be asked (and answered), thus leading to a first informal definition of the adaptation process. The *classification* of AH methods and techniques is outlined by the solid lines representing the typical dimensions for the analysis of adaptive systems [12]; at the same time we join the same classification blocks considering the adaptation *process perspective* which is depicted by dotted line. This process is usually initiated by the user stating the adaptation goal and thus answering the 'Why adaptation is needed?' question. Then in the process we consider the 'What?' and 'To What?' questions, which emphasize Domain Model (DM) and UM descriptions. 'When?' and 'Where?' go next providing context and application area definitions. Lastly, the 'How?' question describes methods and techniques on a conceptual and implementation level and finally all together result in AHS description.

Taking into account user needs and system components (anticipating both core and optionally available components) we would like to present the process which explains the transitions, states, sequences and flows in a generic AHS. First we revisit a few such systems. Then, based on the research and summarization done in [9], we present the flowcharts of an adaptive system and finally come up with the conceptual sequence chart of a layer-structured *Generic Adaptation Framework (GAF)*.

Considering the adaptation process in other systems we mention a few examples of how the authors tried to catch an idea of defining the adaptation processes (both implicitly or explicitly) in their systems and matching processes with the layered structure of their systems.

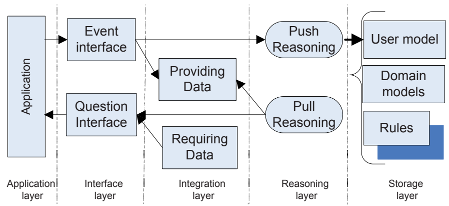


Fig. 4. Overview of the General Ontological Model for Adaptive Web Environments (GOMAWE)

In the GOMAWE system [1] (Fig. 4)the authors tried to fit the adaptation process in the general ontological model of the system they designed. Though there is still much to be considered in terms of the real inter-layer transitions, we can already observe a few basic transitions such as the Event Interface which either triggers the *Push Reasoning* or provides the data for the *Pull Reasoning* interfaces of the Reasoning layer. Here 'Push' is responsible for transforming user events into UM updates which happens when users interact with the system, and 'Pull' retrieves the UM state. Moreover these connections tie different layers of the designed system together.

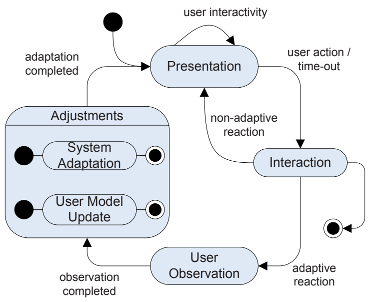


Fig. 5. Lifecycle Model of Adaptation (Munich Model)

The Munich model defined by Koch in [10] (Fig. 5) presented the lifecycle model of adaptation in the UML formalized notion. It defines the following 'layers' or components or states to be tied by these process loops: presentation, interaction, user observation, and adjustments of the systems (which include Adaptation itself and UM updates). These cycles start with an initial presentation and a default UM. Stereotypes are usually used to provide the information for the initial UM. Then the following steps of adaptation cycle follow [10]:

- *System Interaction* - which describes how to react to certain user action(s), resulting in the termination of this cycle and adaptive continuation.
- *User observation* - in which the evaluation of the information got from UM is being done.
- *Adjustments* - comprising the two sub-states: *User model update* - in which UM attributes are updated; *System adaptation* - in which the adaptation is performed (adaptation of presentation, content or navigation) utilizing the state of UM.
- *Presentation* - when the system presents the adaptable elements taking into account the information system knows about the user and remains in this state until the user starts interacting with system over again.

To some extent most of the adaptation 'loops' fall under this classification. Most of these interactions are continuous and recursive when the user continues using the system and explores the knowledge base in depth. We should also mention that here we don't consider any concurrent loops that may happen and influence each other in every aspect.

3 Adaptation Process Flowcharts

In the following section we summarise the procedural knowledge of the data/control and other flows in AHS and come up with the generic representation of AHS processes.

Hereafter we present the adaptation process flowcharts, which generalize the functionality of the AHS. In fact these flowcharts follow the system properties summarization presented in [9], (Tables 1, 2). Based on the summarised (and generalized) functionality we devise these generic adaptation process flowcharts. The abstract representation of the process is shown in Fig. 6 which is elaborated further.

We distinguish the following flowcharts:

- abstract adaptive process flowchart (Fig. 6);
- goal acquisition and adaptation (Fig. 7);
- adaptation functionality (Fig. 8);
- test-feedback functionality.

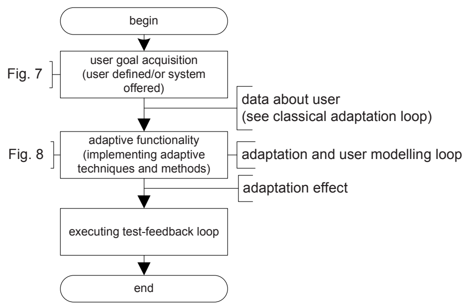


Fig. 6. Generic Adaptation Flowchart (to be considered as the aggregation of Figs. 7-8)

Each flowchart represents a certain aspect of the adaptation process, annotated to give more insight in the description of some blocks. On the right side of each chart we link parts of the process to the layers of the GAF model. The communication between the layers is illustrated in Fig. 9. We also mark with numbers the exact correspondence of Fig. 9 calls and transitions with the outlined blocks on the flowcharts (Figs. 7-8) in order to show the conformity of the sequence and flowchart approaches.

In the *'Goal acquisition and adaptation flowchart' (Fig. 7)* we start with the group analysis, thus assigning the user to a group or acquiring group properties in order to take them into account while choosing the adaptation goals. Here we also make assumptions that the user can belong to only one group and may not switch to another group within a session. The user may have his/her own goal or be advised by the system to (also) use the group goal. In any case goal suitability is checked to determine whether the user can follow it. All suitable goals are elaborated in a sequence of concepts or the most appropriate 'project' (defined set of concepts to study) is chosen.

The *'Adaptation functionality flowchart' (Fig. 8)* presents the main Adaptation Engine (AE) functionality in a sequence of concept-content adaptation steps for a particular user. In general we analyse conditions for a particular step and execute adaptation rules which apply adaptation techniques and perform presentation, content and navigation adaptation. After that UM attributes are updated accordingly and the user proceeds with the next concept either on a 'one-per-click' or project-organized basis. This figure looks very similar to what was done the in IMMPS model [2], presenting a reference architecture for intelligent multimedia presentation systems where the knowledge server was separated from the main flowchart in order to separate and retain the knowledge base from other system functionality. For the same reasons to separate AE functionality we have the distinguished 'concept-content adaptation interaction' block.

'Test-feedback functionality' goes in the end. Here if such a feedback is required the user continues either with the external evaluation or internal assessment which could be the part of a project or a separate questionary or test instance. If this test is failed, user goals might be refined and he/she could be requested start all over again.

4 'Rotating' the Layers of AHS: Adaptation Process and the Layered Model

The conceptual structure of GAF [8] aligns the order of the layers in the system according to the classification of AH methods and techniques (Fig. 3). Though this order represents the basic understanding of the adaptation questions, every particular system may vary or even omit some of these, thus leading to a different composition of the system layers determined by the different adaptation process.

Now, considering the generalized adaptation process flow- charts presented in (Figs. 7-8) and the layered nature of AHS [9] we would like to present the generic adaptation process. We believe that in order to couple, align, sort and arrange the layers of such a system (both generic model or some particular domain focused implementation) one should keep in mind an adaptation process sequence that will partially determine the layers arrangement and to some extent will define the mandatory and optional elements and drive the system design.

Thus we decided to rotate the anticipated layered structure representation by 90 degrees counter-clockwise and match it with the adaptation process flowchart. Fig. 9 shows such an abstraction of a generic adaptation process in terms of the system layers. (It has been rotated once more to fit on the page and be readable.)

We have marked the communication arrows with numbers to set up a correspondence with the flowcharts, where respective blocks are outlined and marked with the same numbers. This is done to show the coherence of the sequence and flowcharts. We should also note that not every connection in the adaptation pro-

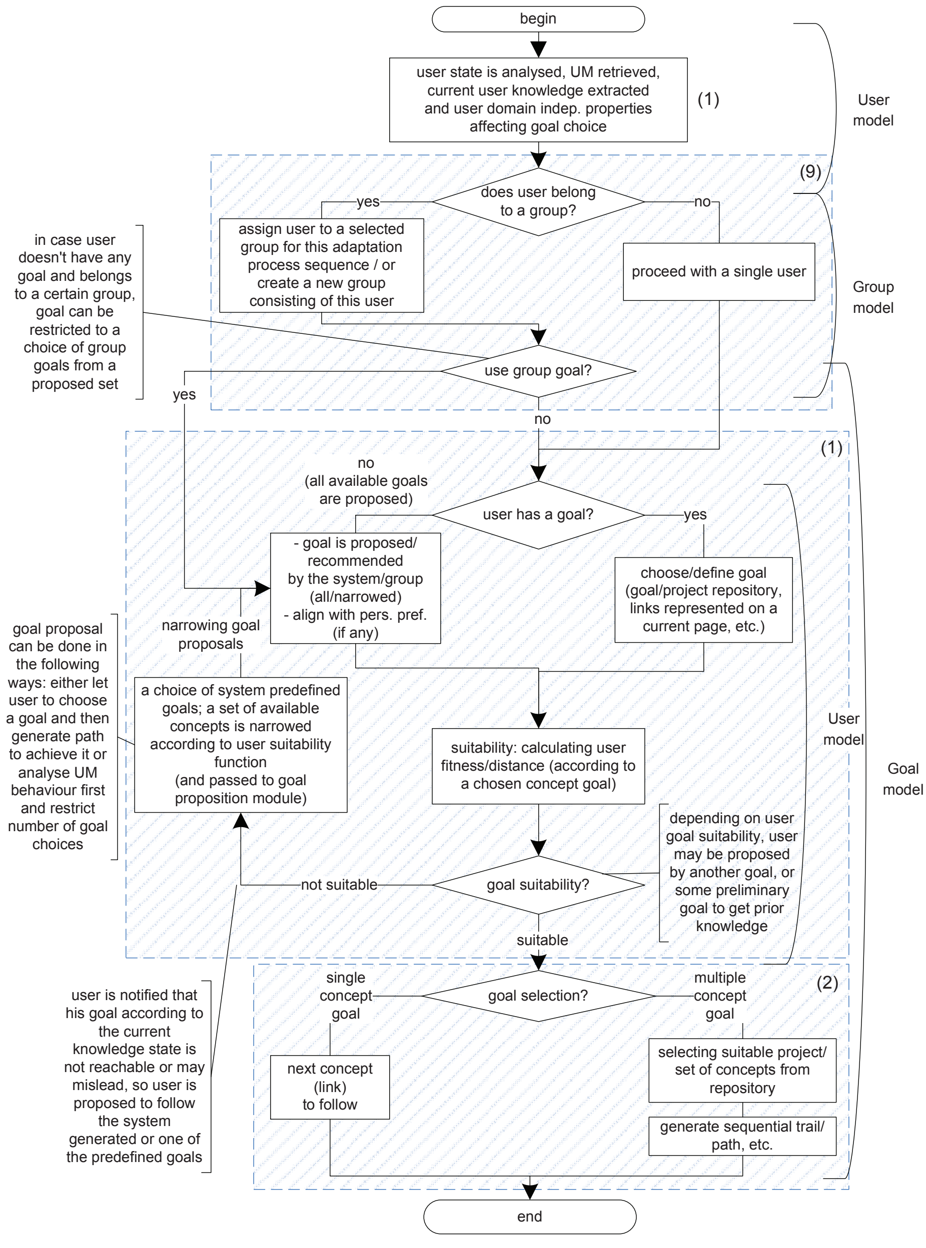


Fig. 7. Goal acquisition and adaptation flowchart

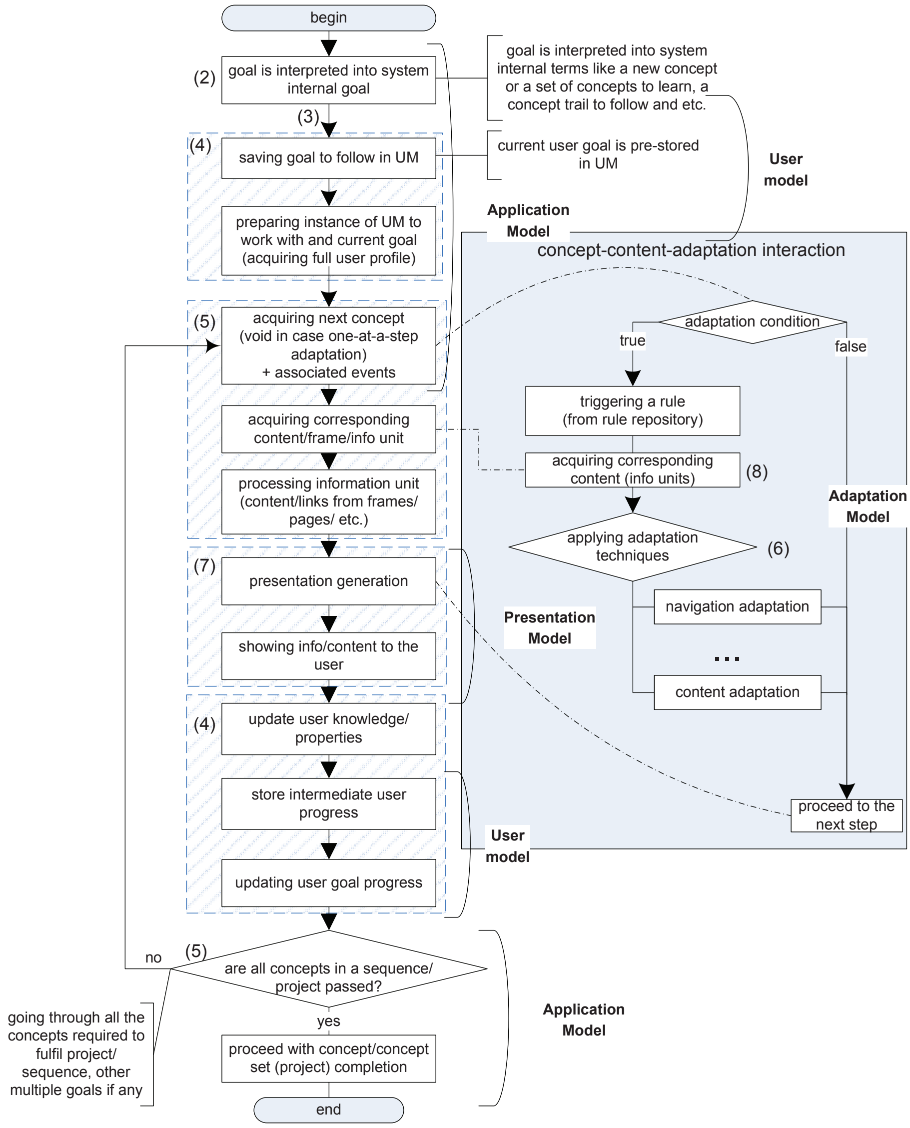


Fig. 8. Adaptation functionality flowchart

cess sequence exists in the above-mentioned flowcharts due to the more extensive description of the GAF process sequence chart. The marked connections are:

1. User goals are defined. In case the user doesn't define any goal it can be proposed by the system or a group goal is used;
2. User goals are aligned with DM, considering the conceptual structure of the domain. According to the selected goal a suitable set of concepts to follow is chosen;
3. Adaptation is initiated and control is passed to the Application Model (AM);
4. Operations of UM properties such as acquisition and update are performed here (corresponds to a few places on the flowchart);
5. Operations mainly concerned working with the concepts from DM;
6. Appropriate adaptation methods and techniques are invoked;
7. Retrieved content is passed to the Presentation model to be rendered/generated and presented to the user;
8. Corresponding content (for concerned concepts) is retrieved from the Resource model and handed over back to AM;
9. Group related operations (assigning users, retrieving group properties, defining new groups, etc.).

5 Conclusions and Future Work

In this paper we defined and elaborated various aspects of the *Generic Adaptation Process*, introduced its model, flowchart and sequence chart. To comply with the layered model we anticipate that the aforementioned process structure will influence the layered composition of the AHS in such a way that the process defined by the system engineer will partially drive the order of the layers of such AHS and define important inter-layer transitions. At the same time we anticipate that the defined processes together with the reference model (e.g. emerging GAF or well known AHAM) may serve as a foundation for the system design, defining not only the system components but the system 'functionality flow' as well, or even deviate into a separate branch of so-called 'process-driven' architectures in the AH field. Moreover the formalized process driven approach gives more insight on AH development methods and unifies this development approach and system organization as it was first mentioned in [10].

We have started elaboration of the generic layered structure of AHS in [9] and then gave the first look at this kind of layered generic structure in [8]. This led to a process-oriented view of a generic layered AHS which was presented in this paper. Based on the research done in [9] we managed to devise a generic flowchart that fits most popular AHS. And finally considering the layered (de)composition of an adaptive system we present a conceptual view of a *generic adaptation process* (Fig. 9).

In the future we plan to extend the adaptation process sequence with more details, elaborate the process description, particularly inter-layer interaction, sustaining the generic approach, at the same time emphasizing the interoperability

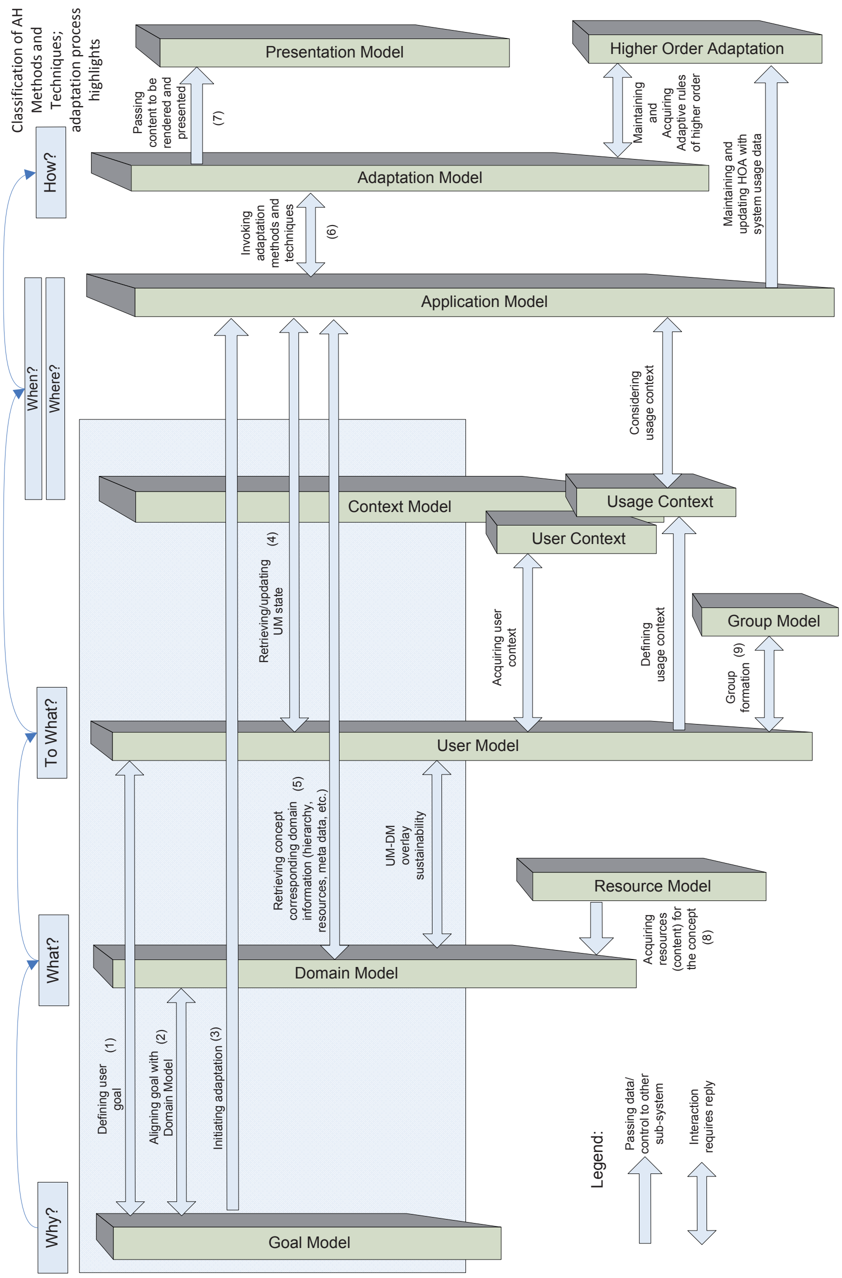


Fig. 9. Conceptual Generic Adaptation Process sequence chart

of a new AH developments (Ontologies, Open Corpus, Higher-Order Adaptation etc.) and the conventional AH approaches. This may lead to describing interoperable and alternative interaction in the system thus representing a generic view of an AH framework which includes all possible variations of adaptation functionality and techniques.

Acknowledgements This work has been supported by the NWO project **GAF**: Generic Adaptation Framework.

References

1. M. Balík and I. Jelínek. Towards semantic web-based adaptive hypermedia model. 2007.
2. M. Bordegoni, G. Faconti, S. Feiner, M. T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A standard reference model for intelligent multimedia presentation systems. *Comput. Stand. Interfaces*, 18(6-7):477–496, 1997.
3. P. Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User Adapted Interaction*, 6(1-2):87–129, 1996.
4. O. Conlan and V. P. Wade. Evaluation of APeLS - an adaptive elearning service based on the multi-model, metadata-driven approach. In *AH*, volume 3137 of *Lecture Notes in Computer Science*, pages 291–295. Springer, 2004.
5. P. De Bra, G.-J. Houben, and Y. Kornatzky. An Extensible Data Model for Hyperdocuments. In *ECHT 92, Proc. of the 4th ACM Conference on Hypertext*, pages 222–231. ACM, 1992.
6. P. De Bra, G.-J. Houben, and H. Wu. AHAM: a Dexter-based reference model for adaptive hypermedia. In *Hypertext '99: Proceedings of the tenth ACM Conference on Hypertext and Hypermedia : returning to our diverse roots*, pages 147–156, New York, NY, USA, 1999. ACM.
7. F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Commun. ACM*, 37(2):30–39, 1994.
8. E. Knutov. GAF: Generic Adaptation Framework. Hypertext 2009 SRC poster, 2009.
9. E. Knutov, P. De Bra, and M. Pechenizkiy. AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Rev. Hypermedia Multimedia*, 15(1):5–38, 2009.
10. N. Koch. *Software engineering for adaptive hypermedia systems*. PhD thesis, Ludwig-Maximilians University of Munich, Munich, Germany, 2001.
11. M. Saksena and A. Cristea. Towards more efficient generic semantic authoring for adaptive hypermedia. In *APS '06: Proceedings of the joint international workshop on Adaptivity, personalization & the semantic web*, pages 11–20, New York, NY, USA, 2006. ACM.
12. M. Schneider-Hufschmidt, T. Ktihme, and M. U., editors. *State of the Art in Adaptive User Interfaces*, pages 13–48. North Holland, Amsterdam, the Netherlands, 1993.

Building Blocks for User Modeling with data from the Social Web

Fabian Abel[1], Nicola Henze[1], Eelco Herder[1], Geert-Jan Houben[2], Daniel Krause[1], Erwin Leonardi[2]

[1] L3S Research Center, Leibniz University Hannover, Germany
{abel,henze,herder,krause}@L3S.de
[2] Web Information Systems, TU Delft, The Netherlands
{g.j.p.m.houben,e.leonardi}@tudelft.nl

Abstract. In this paper we present the so-called Grapple User Modeling Framework (GUMF) which provides user modeling functionality via the Web to applications that would like to offer personalization and adaptation features to their customers. GUMF introduces the notion *dataspaces* which create a logical view on (possibly distributed) user data and provide advanced contextualization and reasoning mechanisms. We showcase the Mypes service that exploits GUMF dataspaces to connect, aggregate, align and enrich user profile information form the Social Web.

1 Introduction

Adaptive systems require information about their users to adapt their functionality [1]. Today, users leave a plethora of traces on the Web that could possibly serve as input for these systems: people provide profile information in social networking services such as Facebook or LinkedIn, annotate their pictures and bookmarks at Flickr or Delicious, and write about their interests in their (micro)blogs. Thereby a lot of useful profile information becomes publicly available [2] . Further, the aggregation of such distributed user data is supported by initiatives such as the Linked Data initiative [3], standardization of APIs (e.g. OpenSocial[1]) and authentication and authorization protocols (e.g. OpenID, OAuth), as well as by (Semantic) Web standards such as RDF, RSS and specific vocabularies such as FOAF[2], SIOC[3], or GUMO [4]. Generic user modeling servers such as CUMULATE [5] or PersonIs [6] facilitate handling of aggregated user data. The Grapple User Modeling Framework (GUMF) [7] follows these approaches and offers means to deliver customized user modeling functionality to its clients.

In this paper we discuss so-called GUMF *dataspaces* that embody and enforce the actual user modeling intelligence of GUMF. We present the *Mypes* service[4] which exploits dataspaces to connect, aggregate, align and enrich user

[1] http://code.google.com/apis/opensocial/
[2] http://xmlns.com/foaf/0.1/
[3] http://rdfs.org/sioc/spec/
[4] http://mypes.groupme.org/

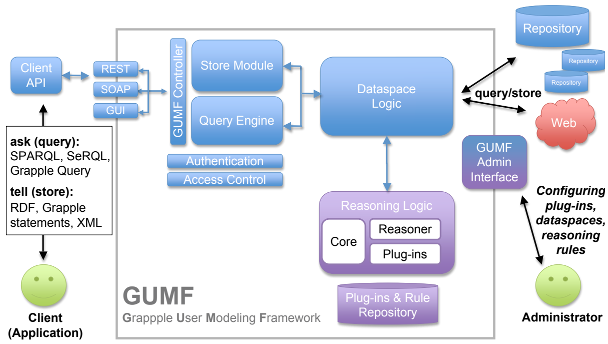


Fig. 1. Architecture of the Grapple User Modeling Framework (GUMF).

profile information form social networking services (Facebook, LinkedIn), social media services (Flickr, Delicious, StumbleUpon, Twitter, Blogspot) and others (Google). Mypes enables users to overview their distributed profiles and via GUMF it allows application developers to integrate aggregated, semantically enriched profile data in RDF or vCard format via a lightweight REST interface into their applications to provide personalization and adaptation functionalities.

2 Grapple User Modeling Framework

Figure 1 shows the architecture of GUMF. The elements at the top provide the essential, generic functionality of the framework; elements part at the bottom right provide generic as well as domain-specific *reasoning logic*. Client applications can access GUMF either via a RESTful or SOAP-based API. Further, there is a *Java Client API* that facilitates development of GUMF client applications[5]. Client applications mainly approach GUMF to store user information (handled by the *Store Module*) or to query for information (handled by *Query Engine*). User profile information is modeled by Grapple statements [7], which are basically reified RDF statements about a user, enriched with provenance metadata. GUMF currently supports SPARQL and SeRQL queries as well as *Grapple query*, a pattern-based query language that exploits the Grapple statement structure to specify what kind of statements should be returned by GUMF. Authorized client requests are answered by GUMF's *Dataspace Logic*. Dataspaces concentrate the actual user modeling capabilities of GUMF. They are equipped with data storage repositories that either reside at the GUMF server or are distributed across the Web (possibly maintained by the client application itself), and with (reasoning) plug-ins that further enrich the data that is available in the repositories or transform user data into a structure/format that is appropriate for the GUMF client applications that request user data.

The *Administrator* of a GUMF client application can configure dataspaces and plug-ins via the *GUMF Admin Interface*. Activating or deactivating plug-

[5] cf. GUMF help pages: http://pcwin530.win.tue.nl:8080/grapple-umf/help/

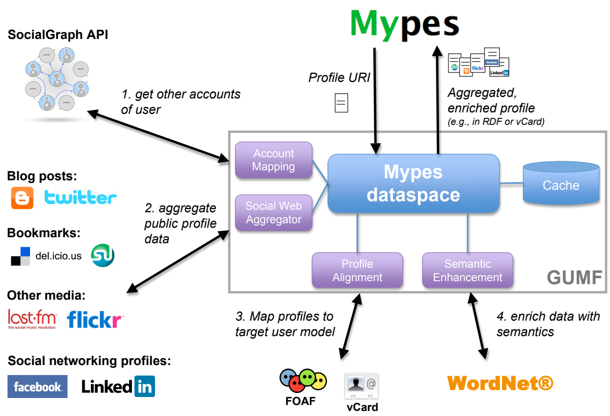


Fig. 2. Aggregation and enrichment of profile data with Mypes.

ins directly influences the behavior of dataspaces. Further, administrators can adjust the plug-ins and reasoning rules to their needs. In the next section we will highlight GUMF dataspaces and describe plug-ins we implemented that support aggregation of user data from the Social Web. These plug-ins and the corresponding dataspace enable Mypes to visualize the distributed user data traces. Moreover, as GUMF dataspaces can be shared across different client applications, we also enable other client applications to benefit from the profile aggregation.

3 Mypes: Enriching User Profiles via GUMF

To illustrate the functionality of GUMF dataspaces we present the Mypes service which is a GUMF client application that particularly highlights GUMF's ability to support the task of gathering information about users for user adaptive systems [1]. In this section we first present the GUMF components we implemented to allow for profile aggregation and enrichment before we describe Mypes features in more detail.

3.1 Linkage, Aggregation, Alignment and Enrichment of User Data

The Grapple User Modeling Framework aims to provide a uniform interface to user data that might be distributed on the Web. To feature access to distributed user data and to align and enhance the data, GUMF and the corresponding GUMF components depicted in Figure 2 respectively perform the following steps.

1. **Account Mapping** Given a user's URI of an online account, the account mapping plug-in gathers other online accounts of the same user by exploiting the Google Social Graph API[6], which provides such mappings for all users who linked their accounts via their Google profile, for example:

[6] http://code.google.com/apis/socialgraph/

traditional profile attributes	Facebook	LinkedIn	Twitter	Blogspot	Flickr	Delicious	Stumble Upon	Last.fm	Google
nickname	x	x	x	x	x	x	x	x	x
first name	x	x							
last name	x	x							
full name	x	x	x		x				x
profile photo	x		x		x				x
about		x							x
email (hash)	x				x				
homepage	x	x	x						x
blog/feed			x	x	x	x	x	x	
location		x	x		x				x
locale settings	x								
interests		x							
education		x							
affiliations	x	x							
industry		x							
tag-based profile					x	x	x	x	
posts			x	x	x	x	x		

Table 1. Profile data for which Mypes provides crawling capabilities: (i) traditional profile attributes, (ii) tag-based profiles (= tagging activities performed by the user), and (iii) blog, photo, and bookmark posts respectively.

```
"http://www.google.com/profiles/fabian.abel": "claimed_nodes": [
    "http://delicious.com/fabianabel",
    "http://fabianabel.stumbleupon.com",
    "http://www.last.fm/user/fabianabel/", ...]
```

For those users where no mappings can be obtained via API, it is possible to provide appropriate mappings by hand. The account mapping module finally provides a list of online accounts that are associated to a particular user.

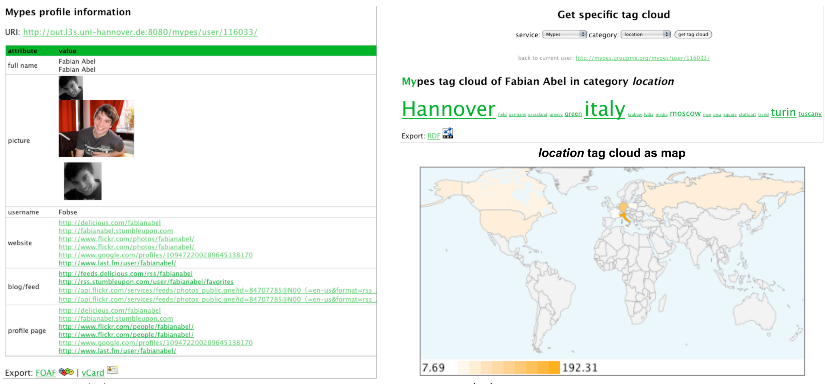
2. **Social Web Aggregator** For the URIs associated to the user, the aggregator module gathers profile data from the corresponding services. In particular, traditional profile information (e.g., name, homepage, location, etc.), tag-based profiles (tagging activities), and posts (e..g, bookmark postings, blog posts, picture uploads) are harvested from nine different services as depicted in Table 1.
3. **Profile Alignment** The profiles gathered from the different services are aligned with GUMF's uniform user model by means of hand-crafted rules, i.e. the user data is modeled by means of Grapple statements [7] using FOAF or vCard as domain-specific vocabulary for the actual user attributes (e.g., name, homepage, etc.). For example, given the *full name* of a user's Google or Flickr profile, GUMF creates a statement as follows.

```
@prefix gc: <http://grapple-project.org/grapple-core/> .
<http://grapple-project.org/gumf/statement-2010-05-19-bob-name>
  rdf:type     gc:Statement;
  gc:user      <http://bob.myopenid.com>;
  gc:predicate <http://xmlns.com/foaf/0.1/name>;
  gc:object    "Bob Mayer";
  gc:created   "2010-05-19T16:23:04.243+01:00" .
```

4. **Semantic Enrichment** Tag-based profiles are further enriched and clustered by means of WordNet[7] categories so that GUMF client applications can, for example, access particular parts of a tag-based profile such as facets related to *locations* or *people*.

The four plug-ins can be plugged into dataspaces. We applied them to the Mypes dataspace which forms the basis for the Mypes service.

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



(a) traditional profile (b) filtering tag-based profile

Fig. 3. Aggregation of traditional and tag-based profiles.

3.2 Mypes Service

Mypes exploits GUMF to provide an interface that is valuable for casual users, who would like to overview their distributed profile data. Further, it makes the enriched public profile data that is available via the Mypes dataspace (see Figure 2) also available via a lightweight RESTful interface (in FOAF and vCard format). Hence, thanks to GUMF and the plug-ins presented above, Mypes just has to focus on representing the public profile data – either for end-users or for systems that require additional information about their users.

Given the Google profile URI of a user, Mypes queries GUMF and the Mypes dataspace particularly to obtain the corresponding aggregated, enriched profile data. It then enables users to overview their public profiles, for example, users can inspect particular profile values, analyze the completes of their profiles on a bar chart and understand to what kind of information about themselves is publicly available. Mypes thus raises the users' awareness of their profile data distributed across different services on the Web.

Figure 3(a) shows an example of an aggregated Mypes profile, namely the traditional profile attributes gathered from the diverse services (see Table 1). When accessing `http://mypes.groupme.org/mypes/user/116033/rdf` the FOAF profile in RDF/XML syntax is returned. Mypes exports all available values for a profile attribute, e.g., if a user specifies her name differently at the different services then all these different values are provided.

GUMF also connects the tagging activities users perform in the different tagging systems. As the *semantic enrichment* plug-in (see Fig. 2) extends tag assignments with meta-information that states to which WordNet category the corresponding tag belongs to, it is possible to filter the aggregated tag cloud of a user according to WordNet categories. For example, Figure 3(b) shows the aggregated tag cloud filtered so that only tags related to locations are displayed. For this kind of tag cloud, Mypes provides an alternative visualization: tags related to locations are mapped to country codes (using the *GeoNames* Web service[8]), which are sent to Google's visualization API to draw a geographical intensity map that highlights those countries that are frequently (possibly indirectly) referenced by tags in the profile (see bottom in Figure 3(b)). Mypes also features RDF export for these (specific facets of) tag-based profiles using the Tag Ontology[9] and SCOT[10] vocabulary.

4 Conclusion

In this paper we described how the Grapple User Modeling Framework (GUMF) enables to enrich user profiles with user data gathered from the Social Web. We presented the Mypes service that exploits this functionality to enable casual users to overview their distributed profiles and allows adaptive applications to re-use the aggregated and enhanced profile information for their own purposes. In our future work we plan to evaluate the actual benefit for end-users of adaptive applications which make their adaptation decisions based on the enriched profiles produced by GUMF and Mypes respectively.

Acknowledgments This work is partially sponsored by the EU FP7 project GRAPPLE (http://www.grapple-project.org/).

References

1. Jameson, A.: Adaptive interfaces and agents. The HCI handbook: fundamentals, evolving technologies and emerging applications (2003) 305–330
2. Abel, F., Henze, N., Herder, E., Krause, D.: Interweaving public user profiles on the web. In Kobsa, A., Bra, P.D., eds.: International Conference on User Modeling, Adaptation and Personalization (UMAP), Hawaii, USA, Springer (June 2010)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) **5**(3) (2009) 1–22
4. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo - the general user model ontology. In Ardissono, L., Brna, P., Mitrovic, A., eds.: User Modeling. Volume 3538 of Lecture Notes in Computer Science., Springer (2005) 428–432
5. Yudelson, M., Brusilovsky, P., Zadorozhny, V.: A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In Conati, C., McCoy, K.F., Paliouras, G., eds.: User Modeling. Volume 4511 of Lecture Notes in Computer Science., Springer (2007) 27–36
6. Assad, M., Carmichael, D., Kay, J., Kummerfeld, B.: PersonisAD: Distributed, active, scrutable model framework for context-aware services. In LaMarca, A., Langheinrich, M., Truong, K.N., eds.: Pervasive Computing. Volume 4480 of Lecture Notes in Computer Science., Springer (2007) 55–72
7. Abel, F., Heckmann, D., Herder, E., Hidders, J., Houben, G.J., Krause, D., Leonardi, E., van der Slujis, K.: A framework for flexible user profile mashups. In Dattolo, A., Tasso, C., Farzan, R., Kleanthous, S., Vallejo, D.B., Vassileva, J., eds.: Int. Workshop on Adaptation and Personalization for Web 2.0 at UMAP '09, CEUR Workshop Proceedings (2009) 1–10

[8] http://www.geonames.org/
[9] http://www.holygoat.co.uk/projects/tags/
[10] http://scot-project.org/scot/

Towards a Generic Platform for Indoor Positioning using Existing Infrastructure and Symbolic Maps

Kurt Gubi[1], Rainer Wasinger[1], Michael Fry[1], Judy Kay[1], Tsvi Kuflik[2], Bob Kummerfeld[1]

[1] CHAI Group, School of IT, Sydney University, 2006, NSW, Australia
kurt@it.usyd.edu.au, {rainer.wasinger, michael.fry, judy.kay, bob.kummerfeld}@sydney.edu.au
[2] University of Haifa, Haifa, Israel
tsvikak@is.haifa.ac.il

Abstract. One of the important challenges for personalised, context-aware information delivery within buildings is to be able to show the user a map with their own location as well as the locations of points of interest for them. One very desirable property of a personalised, context-aware mobile application is that it can operate at the same time as preserving the user's privacy. To achieve this, we need to do on-device positioning and personalisation. This paper presents the design of a platform, and its implementation for the retrieval of publicly available building data (symbolic maps and associated radio-frequency infrastructure point locations) for the purpose of coarse-grained indoor positioning on mobile devices. In comparison to other indoor positioning systems, this work focuses on the mechanism through which building data is made available to mobile devices, as too the motivation in providing generic coarse-grained indoor positioning based on the use of existing infrastructure and building data.

Keywords: Indoor positioning; symbolic maps; semi-unprepared environments; client-side personalisation

1 Introduction and Motivation

Location-aware services are becoming increasingly common. A key reason for this is the growing number of mobile devices that can now determine their location and have the computational and communication power to deliver sophisticated services. Examples of such devices are smartphones, eReaders, mobile gaming consoles, in-car consoles, and netbooks. Most of these devices now come with a variety of inbuilt technologies as standard: for location sensing, e.g. accelerometer, magnetic field, orientation sensors, GPS; for positioning and communication, e.g. radio-frequency (RF) technologies like 3G, WLAN, and Bluetooth; and also for IO, e.g. inbuilt cameras that can be used for vision sensing, particularly in combination with AR and QR tags. High-profile investments, such as the European Galileo satellite system that is currently being built, are an indication of the value that is placed on location-aware services.

Modern positioning (and navigation) systems now cater for a wide range of scenarios ranging from in-car, to on-foot, and both outside and inside of buildings (e.g. [1]). In comparison to outdoor positioning via GPS, indoor positioning is far less widespread, and there are a number of reasons for this. One reason is the lack of availability of suitable maps. It is important to appreciate that the creation and maintenance of indoor maps is inherently different than is the case for outdoor maps. For example, access to - as well as suitability and privacy of - building blueprints provides a barrier to entry, meaning that coverage of indoor spaces, e.g. on the scale of a whole city, is far less accomplished than is the case for outdoor map locations. This means that we need to explore a different form of mapping approach for indoor positioning, and this must be able to operate with the type of indoor maps that are widely available; notably, we believe it must make use of symbolic maps that are often already available for buildings, despite these often not being particularly accurate in terms of scale and these often being highly selective in the information shown on the map.

Another key challenge of indoor positioning follows from the limitations of the technologies available. While there are several existing and emerging technologies that have been used in indoor positioning prototypes (e.g. based on RF, visual technology, dead-reckoning techniques), each of these has its own merits and pitfalls. None of them, taken independently, have the same planetary-scale applicability, nor the consistent accuracy that GPS provides for outdoor positioning. This means that there is still important research to be done to create systems that can make use of a combination of the available location technologies to achieve effective indoor positioning of a person as they move around a building. Finally, specialty-built indoor positioning solutions often require infrastructural (and software) outlays that are not always feasible.

In this paper, we describe our design for a platform that addresses these problems and describe its implementation for the retrieval of “publicly” available building data in the form of symbolic maps and markup of associated RF infrastructure point locations (though not limited to just RF) like WLAN and Bluetooth.

It can be noted that such a platform and its associated APIs will be an indispensible building block for web-based user-adaptive systems that contain any type of indoor positioning component.

In Section 2, we describe the benefits for both providers and users of such a platform. Section 3 provides an overview of the platform and details of the implemented proof-of-concept client application for Android smartphones. This is followed in Section 4 with a summary of related work in the fields of positioning platforms, and symbolic maps and data modelling. The paper concludes with a description of future work in Section 5.

2 Indoor positioning in Semi-unprepared Building Environments

Buildings are typically constructed based on highly accurate geometric blueprints, which although useful for architects and builders, are rarely accessible and rarely relevant (with regards to the detail they show) to general visitors of the building. Many 'public' buildings (i.e. buildings that the general public have access to, either with or without entrance costs attached) do however have maps available to the public (e.g. consider museums, libraries, theatres, hospitals, and so on). These maps are symbolic in nature, meaning that they need not align to any geometric model or linear scale, but instead are specifically designed to highlight aspects deemed to be most relevant to the user.

Similarly, many buildings are nowadays fitted with a range of RF-based communication technologies like Bluetooth and WLAN, and although building administrators may be reluctant to add additional infrastructure specifically for the purpose of indoor positioning, the modelling of already existing infrastructure may be an acceptable compromise. We call such environments "semi-unprepared" in that no additional technologies need be integrated, but the modelling of existing sensor/beacon points is still required.

Consider the following scenario. Tom, a tourist, is keen to visit a well known local museum. Upon arriving at the museum, he loads up the RoughMaps application on his smartphone and is presented with a number of icons on his screen representing nearby public buildings (e.g. museums, libraries, shopping malls). After Tom has selected the particular museum of interest, RoughMaps downloads the relevant mapping data from the web-service via a http request, and presents Tom with a number of symbolic maps, each one typically showing one level in the museum. While Tom considers these maps a useful feature, he is unsure of where he is in the building, so he presses the "Find Me" menu item, and the system positions him on the relevant sub-map. He is also able to take a photo of any of the QR codes scattered around the museum to have his position updated on the map. As he walks around, his position is updated on the map through the use of a dead-reckoning approach that combines readings from the digital compass and accelerometer sensor (i.e. a directional pedometer) contained within his Nexus One smartphone.

This scenario describes how a mobile client-device accesses (through a web service and its associated set of APIs) public indoor map data, to provide an end-user with symbolic indoor maps and indoor positioning information. Such a service would enable different mobile device types (including the myriad of smartphones) to provide personalised context-aware information relating to individual building spaces. Some of the indoor-based context-aware applications that such a service would enable include: personalised tour guides, recommendations for paths to follow and POIs to see (e.g. based on crowd-sourced data), detailed information pop-ups on nearby and relevant POIs, and educational treasure-hunt games for exploring indoor spaces.

3 Server-side Platform and Client-side Demonstrator

This section describes the platform through which building data is made available to mobile clients and the proof-of-concept client application for Android smartphones.

There are two main components described in the above scenario, namely: a web-service that allows for the ‘publishing’ of symbolic map data and associated sensor location points; and an API/client-interface that allows for such map data to be downloaded and interpreted by mobile applications (and foreseeably also web clients in the future). These components are shown in Figure 1A. Figure 1B shows the interface in which building data in the form of floor plans and sensor/beacon-location markup can be published to the server, while Figure’s 1C and 1D show how such map data is selected and downloaded by the user from a client device. It should be noted that certain complexities have been left out of the client-side implementation thus far; in particular, the client-side application only uses QR codes and dead-reckoning to provide indoor positioning information back to the user. This implementation is however clearly extensible to the sensing of additional beacons such as those based on RF technology, and the overarching mechanisms in which other applications and web-services are able to access the symbolic map data are also left unaffected from client-side implementations.

Fig. 1. Client-server architecture (A), the web-interface allowing building administrators to publish building data (B), and map data being selected and downloaded by the end user (C, D).

The generic indoor positioning component described in the scenario above is relevant to a broad range of mobile systems. For example, in [2], a subset of mobile systems are described, namely adaptive mobile guides, and it can be noted that all of the systems described in that work, ranging from museum guides and navigation systems to shopping assistants, use location as part of their application context.

4 Related Work

This work most closely relates to the intersecting fields of indoor positioning platforms, symbolic map use, and data modelling techniques for indoor spaces.

Indoor Positioning Platforms: A number of indoor positioning platforms have been created over the past two decades. The Active Badge system (1992) [3], MIT's Cricket system (2000) [4], BlueStar (2004) [5], and the Personal Navigator (2004) [6] are important examples of such systems.

The Active Badge system represents a class of indoor positioning system in which end-users are required to wear tags that broadcast their location to a centralized service through a network of sensors. The Cricket system, in contrast, represents the class of indoor positioning systems that are based on a decentralized approach, which has the particularly important property of being privacy preserving. In this case, the user carries a specially-designed listening device, which estimates its distance from nearby positioning beacons. The BlueStar and Personal Navigator systems take this basic idea further by allowing the client-side 'location-sniffing' device to be an off-the-shelf commodity phone and/or PDA. Given the importance of location privacy, we have taken a similar location-sniffing approach to BlueStar and Personal Navigator. We move beyond the previous work in that we make use of a range of facilities that are available on the user's smartphone, with various APIs to allow for generic implementation by any number of 3rd-party applications designed for mobile client devices (and foreseeably also mobile web services).

Symbolic Map Use and Data Modelling Techniques for Indoor Spaces: Research into human cognition has identified the use of landmarks for positioning and navigation as immensely useful. In [7], a number of papers are surveyed in which the importance of human conception of space as a collection of familiar landmarks has been shown both behaviourally (e.g. for newcomers to a city) and cognitively. Indeed in [8], it is described how human cognitive maps - by their very nature of needing to find a balance between storing as much useful information as possible against the need to keep the amount of information at a manageable level - emphasise some information at the expense of other data.

Tourist maps, for example, are quite often symbolic in nature, and this is often done to increase the salience of map features that are deemed relevant to the viewer, at the cost of decreasing the salience of the remaining map features/detail. It is this form of graphical symbolic map, which quite often bears little resemblance to the geometric blueprints of the buildings they represent, that we place at the heart of this work and its associated server-side platform and client-side demonstrator.

The Yamamoto map modelling toolkit [9] is one solution that can be used for the modelling of indoor spaces. Yamamoto provides support for the geometric modelling of architectural ground plans through polygon meshes. It is a desktop application written in C# for the .NET framework and has many features that would make it an ideal tool to use, though does not currently offer its functionality in the form of a web service, and would thus require users wishing to upload map data to first download and install the toolkit. Yamamoto also does not focus specifically on the modelling of symbolic maps that may bear little resemblance to their associated geometric building blueprints.

5 Conclusions and Future Work

This paper provides a number of outcomes. Firstly, it describes a platform that allows for single-point of access for downloading publically available indoor mapping data. Secondly, it provides the mechanism in which sensor/beacon location information can be utilised for coarse-grained indoor positioning. Thirdly, it makes use of a developed API and a sample client-side Android implementation of those APIs to demonstrate, as proof-of-concept, how to use the platform.

Future work will focus on continued implementation of the markup notation used to model infrastructure points; surveys into the level of infrastructure that different types of public buildings currently contain; and usability studies into what minimal level of accuracy is required for indoor positioning to be considered useful by end users.

This work is funded by the Smart Services CRC, as part of the Multi-channel Content Delivery and Mobile Personalisation Project. We would also like to acknowledge the Australian Museum whose floor plans are shown in this paper.

References

1. Krüger, A., Butz, A., Müller, C., Stahl, C., Wasinger, R., Steinberg, K-E., Dirschl, A., The Connected User Interface: Realizing a Personal Situated Navigation Service. In: Proc. of the 9th International Conference on Intelligent User Interfaces, pp. 161 – 168 (2004)
2. Krüger, A., Baus, J., Heckmann, D., Kruppa, M., Wasinger, R., Adaptive Mobile Guides. Chapter in: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, pp. 521 – 549, Springer (2007)
3. Want, R., Hopper, A., Falcão, V., Gibbons, J., The Active Badge location system. ACM Transactions on Information Systems, 10 (1), pp. 91 – 102 (1992)
4. Priyantha, N., Chakraborthy, A., Balakrishnan, H., The Cricket Location-Support System, In: Proc. Of the International Conference on Mobile Computing and Networking, pp. 32-43 (2000).
5. Quigley, A., Ward, B., Ottrey, C., Cutting, D., Kummerfeld, R., BlueStar, a privacy centric location aware system. In: IEEE Position, Location and Navigation Symposium, vol. 1 (2004)
6. Wasinger, R., Krüger, A., Multi-modal Interaction with Mobile Navigation Systems. In: W. Wahlster (ed.): Special Journal Issue "Conversational User Interfaces", it - Information Technology, vol 46, number 6, pp. 322 – 331, Oldenbourg (2004)
7. Bidwell, N. J., Lueg, C. P., Creating a Framework for Situated Way-Finding Research. In APCHI, pp. 40 – 49 (2004)
8. Golledge, R. G., Wayfinding behavior: cognitive mapping and other spatial processes, The Johns Hopkins University Press (1999)
9. Stahl, C., Haupert, J., Taking Location Modelling to new Levels: A Map Modelling Toolkit for Intelligent Environments. In: 2nd International Workshop on Location- and Context-Awareness. In: M. Hazas, J. Krumm, and T. Strang (Eds.), pp. 74 – 85, Springer (2006)

Ontology-Driven Adaptive Accessible Interfaces in the INREDIS project

Raul Miñón, Amaia Aizpurua, Idoia Cearreta, Nestor Garay, Julio Abascal

Laboratory of HCI for Special Needs
University of the Basque Country (UPV/EHU)
Informatika Fakultatea. Manuel Lardizabal 1, 20018 Donostia, Spain
{raul.minon, amaia.aizpurua, idoia.cearreta, nestor.garay, julio.abascal}@ehu.es

Abstract. This paper presents and architecture that is being developed to generate adaptive accessible interfaces taking into consideration the features of users with disabilities, the characteristics of the devices that they use to access remote services, and other context features. In order to perform the modelling of the diverse parameters under consideration an ontological approach has been chosen. This research is part of the INREDIS project that has the purpose of creating a universally accessible, interoperable and ubiquitous environment to allow people with disabilities to control service machines and /or different targets.

Keywords: adaptive interfaces, accessibility, ontologies, user modelling.

1 Introduction

Services that are accessible through the Internet should be accessible, ubiquitous, and interoperable. Unfortunately, this is not the current situation since most interfaces do not take into account the needs of the users nor the context in which the interaction is performed. The specific characteristics of the device used to request the service are also frequently ignored, which has a negative impact on the accessibility, usability, ubiquity and interoperability of the service. Adaptation is very helpful to overcome accessibility barriers. It is based on user modelling that allows tailored interactions based on relevant user features such as skills, preferences, interests, etc. In addition, a number of other context features are taken into account to optimize the adaptation process, such as the task and the objective that the user wants to achieve.

The research work presented in this paper is carried out in the framework of the INREDIS project [1], which aims to provide universal accessibility, interoperability and ubiquity in order to allow people with disabilities to control service machines, vending machines, home appliances, etc. In this paper we aim to depict the architecture of the Interface Generator module that is used to produce adaptive interfaces taking into account the relevant characteristics of users with disabilities and context features, including the characteristics of the devices used to access these services. Ontologies are used in order to perform the modelling of the diverse parameters taken into consideration because they allow reusability, besides they allow extracting new information through inferences.

In relation to the method for storing and processing data for modelling, some approaches propose mark-up languages to model the user (e.g. HumanML [2]) while others propose the use of ontologies (such as GUMO [3]). In addition to the user characteristics, other projects (such as CAP [4]), also consider specific issues related to context. Most of these approaches are predominantly focused on user modelling. Though, adaptive interfaces devoted to the support of people with disabilities have to consider several other issues included in the interaction context, such as accessibility [5] [6] or affective resources [7]. Taking into account the principal advantages offered by these researches, our approach is focused on modelling not only the user but also the technological context and non-technological context.

3 The INREDIS Approach

In order to create adaptive accessible interfaces for people with disabilities within the INREDIS project, a module called the "Interface Generator" has been designed. Its aim is to provide universal access for anyone, including people with disabilities, regardless the device used or the service that is accessed. To achieve this objective, it is necessary to model the different aspects of the domain where the user is interacting. Several approaches have been studied to model these features and eventually an ontological model was chosen, since ontologies offer automated reasoning, dynamic classification and consistence checking giving the opportunity of extracting other relevant features without interfere the user.

In order to take into account all relevant aspects of the user and his context, three different ontologies have been designed and some others are still in the conceptualization phase. Subsequently, all the ontologies designed have been integrated through a global ontology that is modular enough to encompass future ontologies, if they are needed. *User ontology* has been designed to model the skills and characteristics of the users. *User Device Ontology* aims to gather all the information about the device used by the user to interact with the system. *Target Ontology* has the objective of modelling the characteristics of the possible targets or services that can be integrated, such as ATMs, vending machines, and so on.

In addition to these ontologies, there are others in the conceptualization phase that will be built when they are required. These are *Interface Ontology* to guide the process of deciding interface mark-up language and components*; Adaptations Ontology* to model the best adaptations for each user stereotype; and *Assistive Technologies Ontology* to model the diverse assistive technologies accessible trough the INREDIS framework. More information can be obtained in [8].

4 Proposed architecture and component modules

The architecture designed to generate adaptive accessible interfaces is composed of diverse modules, each one provided with specific functionalities to perform specific tasks. It is highly flexible to enable the incorporation of new modules providing new functionalities in the future. *IG Manager Module* is an orchestrator that manages the whole process and the functionalities of the Interface Generator. *Constructor Module* is responsible for creating a default interface in an abstract language. *Resources Manager Module* analyses the resources provided by the target to determine whether these resources are compatible with the user and the user's device. *Selector Module* has to decide which adaptations must be performed to obtain an accessible interface. *Adapter Module* applies the adaptations selected by the selector module to the default interface, obtaining an adaptive accessible interface, supported by the user and his/her device and which is able to control target functionalities. *Data Injector Module* checks whether there is an interface previously generated for the current type of user, device and target. More information about these modules can be obtained in [8].

5 Conclusion

A prototype has been developed to verify the adequacy of the architecture proposed. Although the interaction with the ontologies was only simulated, this prototype showed the functionality and validity of each module and detected its inconsistencies and deficiencies. We are currently developing a more complete version of the Interface Generator that extends its features to allow adaptation to all the stereotypes provided by the general ontology, regardless the device and the target.

Acknowledgements. The involved work has received financial support from the Department of Education, Universities and Research of the Basque Government.

6 References

1. INREDIS project: http://www.inredis.es/Default.aspx. Last accessed: 18 May, 2010.
2. Oasis. http://www.oasis-open.org/committees/download.php/60/HM.Primary-Base-Spec-1.0.html. Last accessed: 18 May, 2010.
3. Heckmann, D., et al. GUMO - the General User Model Ontology. Procs. of the 10th Int. Conf. on User Modeling. LNAI 3538, Springer-Verlag. Berlin, 2005. Pp. 428-432.
4. Fourney, D. Using a common accessibility profile to improve accessibility. PhD Thesis, University of Saskatchewan.
5. Vigo, M., Aizpurua, A., Arrue, M. and Abascal, J. Automatic Creation of User Profiles for Achieving Personal Web Accessibility. Procs. of the Int. Conf. on Computers Helping People with Special Needs 2008. LNCS 5105, 354-361. Springer, 2008.
6. Stephanidis, C. (Ed.). The Universal Access Handbook. CRC Taylor & Francis Group. 2009.
7. Cearreta, I. et al. Models to facilitate the development of affective resources. Procs. of Eur. Conf. on Cognitive Ergonomics 2008. Funchal (Portugal), 2008. Pp. 69-70
8. Miñón R. et al. An Architecture for Ontology-Driven Adaptive Accessible Interfaces. Internal Research Report No. EHU-KAT-IK-07-10. http://www.kat.ehu.es/s0139-lipcnec/es/contenidos/informacion/gi0163_publicaciones/es_ 00163_pu/00163_publicaciones.html

PERSEUS – Personalization Services Engine

Michael Yudelson[1]

[1] University of Pittsburgh, School of Information Sciences,
135 North Bellefield Ave. Pittsburgh, PA 15260, USA
mvy3@pitt.edu

Abstract. PERSEUS is a standalone personalization service provider. It follows the idea of separating the adaptation logic from the AHS and offering it as an on-demand service. PERSEUS provides full abstraction from the adaptation models and methods. Since its first deployment, PERSEUS has seen intensive use and has been the primary source of personalization in about 20 course-semesters. It offers a set of adaptive guidance and recommendation services. The target of this demo is to demonstrate PERSEUS's potential of providing personalized access to the interactive educational resources.

Keywords: service-based personalization, adaptation, adaptive hypermedia

1 Introduction

Modularization and component reuse is one of the major and most promising trends in the field of adaptive educational hypermedia. This process has started even before adaptive hypermedia systems came to existence. Early user modeling shells separated from the user-adaptive systems in the 1980s. In the late 1990s, open-corpus hypermedia allowed content to be added at the run-time rather than at the design time. Today we are witnessing the emerging split of the adaptation models and methods themselves form the adaptive hypermedia system (AHS) that changes from all-in-one tool to the open integrator of the interactive technology

In this demo we are presenting PERSEUS, a personalization server that offers an abstraction of the adaptation methods to the AHS developers. With PERSEUS adaptation can be consumed rather than built-in and can be reused in many contexts and easily replaced without changing the structure of the content. PERSEUS reduces the problem of adaptation provision to the problem of configuration. A recently added innovative feature of PERSEUS – an embedlet – allows adaptation to be built even into static HTML page.

2 PERSEUS – Personalization Services Engine

The conceptual idea of how PERSEUS works is shown in Fig.1. A content management system (here Knowledge Tree portal) provides access to a preconstructed hyperspace. To render a personalized view of a particular page, the portal consults the personalization service engine. To do that, the portal sends the structure of the currently viewed page (as an RSS1.0/RDF feed) and context information (user/group id, personalization algorithm code, etc.). PERSEUS queries user modeling server(s) (and/or other data sources known to it) and performs the adaptation that was requested. The returned result is an original RSS1.0/RDF page feed with personalizing updates. The new feed may have original links reordered or removed, new links inserted, annotations added to links. The portal parses the feed and renders a personalized page for the user.

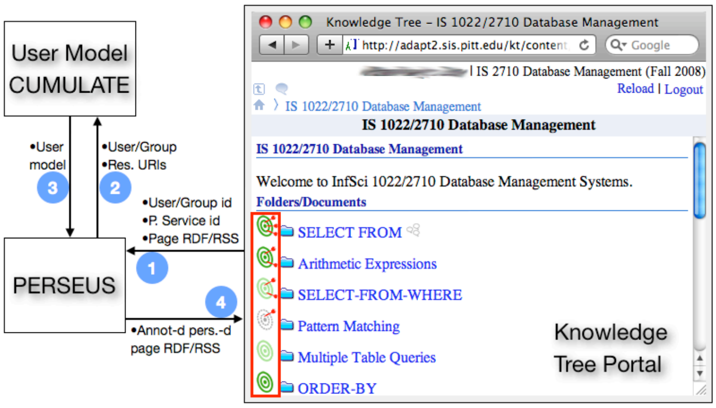


Fig. 1. Example of PERSEUS's topic-based adaptive navigation for Knowledge Tree [1] portal. Adaptive annotations (targets with darts) produced by PERSEUS are enclosed in a square. Here CUMULATE [2] is utilized as the user modeling server.

The compliance threshold for the portal to be able to use PERSEUS is minimal. Every personalization method implemented in PERSEUS is exposed as a RESTful web service. The portal has to be able to package its pages' link structures as a simple RSS1.0/RDF document and send it as one of the parameters to the selected service URL. PERSEUSE's response – modifications to the link structure, including annotations with descriptive JavaScript tooltips – should be parsed again.

3 PERSEUS Embedlets – Adaptation Made Easy

A recent extension to the PERSEUS called *embedlet* allows adaptive hypermedia authors to skip data exchange protocol in its entirety and paste snippets of adaptively annotated lists of resources as plain HTML code. An embedlet is a stationary configured call to one of the PERSEUS's adaptation techniques. It is comprised of an RSS1.0/RDF document, containing a flat link list representing a large portion of the hyperspace and a pointer to the desired personalization service. Each embedlet is exposed as quasi personalization service.

To invoke an embedlet one should insert an object HTML tag into the web page with data attribute pointing to the embedlet's URL. To only show part of the links from the embedlet's exhaustive list one must use an additional parameter and specify an enumeration of links numbers in the bound list. User and/or group identity need to be present in the embedlet as well. However, in the case of group-based navigation (cf. Fig.2), no individual users are distinguished, and group identity could also be statically bound, which allows adaptive navigation embedlets to be successfully used in static HTML pages.

Embedlets are equivalent to regular PERSEUS services in terms of adaptation functionality offered. However, in terms of authoring they are significantly easier to aggregate into existing content.

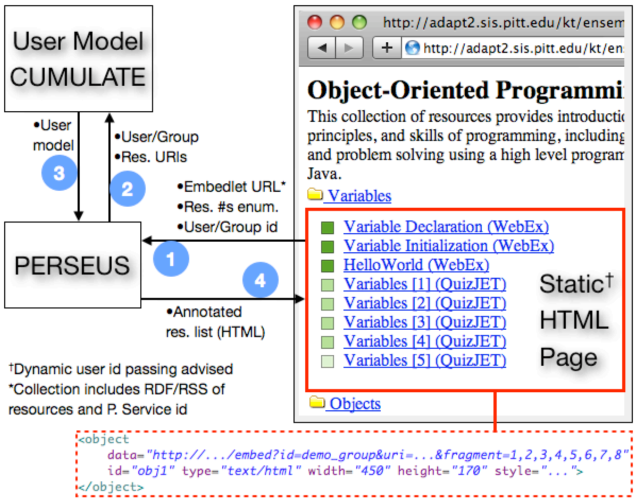


Fig. 2. PERSEUS's group-based social navigation support as an embedlet.

References

1. Brusilovsky, P., Sosnovsky, S. A., Yudelson, M., Lee, D. H., Zadorozhny, V., and Zhou, X. (2010). Learning SQL Programming with Interactive Tools: From Integration to Personalization. ACM Transactions on Computing Education, 9(4), 1-15.
2. Brusilovsky, P., Sosnovsky, S. A., and Shcherbinina, O. (2005). User Modeling in a Distributed E-Learning Architecture. In L. Ardissono, P. Brna, and A. Mitrovic (Eds.), 10th International Conference on User Modeling (UM 2005), (pp. 387-391).