Case-Based to Content-Based User Model Mediation

Shlomo Berkovsky[1], Ariel Gorfinkel[1], Tsvi Kuflik[1] and Larry Manevitz[1]

Abstract. Systems providing personalized services to users have a need to build and maintain a User Model (UM). However, at the onset of providing services, such a system has no prior knowledge about a user and it may benefit from information imported from external sources. Due to lack of standards in the representation of UMs, commercial competition, and privacy issues, distinct personalized service-providing systems build their own specific models and store their information in incompatible manners. Thus, although much data on a specific user might exist in other systems; it is typically unavailable for use in the initial phase of the given system. This work puts forward the design of a user model mediation idea. This is demonstrated in an initial implementation in a specific system (Museum Visitors' guide system) under the PIL project, where the user is modelled by a "bag of words" vector and the initial information is imported from a case-based modelled user (in an external trip planning system).

1 INTRODUCTION

In order to provide users with personalized services, systems build and maintain a UM for each user. In general, a UM may be comprised of various details of personal information such as the user's age, education, income, life style, interests, preferences, past interactions with the system, etc.

Different systems use various methods, and techniques from diverse research areas, such as information retrieval, artificial intelligence and behavioural sciences for the construction of a UM [4]. Furthermore, every system stores UMs according to its own representation and chooses only the specific parts of user data that are relevant for providing its personalized services. Thus, large portions of user-related data that are heterogeneous both in representation and in content are distributed over various systems.

The notion of general (i.e. application-independent) user modeling was initially proposed in [3]. Their system - 'General User Modeling System' (GUMS) allowed the developers of user-adaptive applications to define simple user stereotype hierarchies. GUMS determined the basic functionality of general user modeling systems: providing selected runtime personalized services that can be configured during development time.

Most of the general systems that were developed, classified the collected UM to one of the predefined stereotypes using different inference methods. For example, [7] allowed stereotypical assumptions about the user and users groups to be represented in a first-order predicate logic, so that inferences across different assumptions could be defined in a first-order modal logic.

In [5], the authors discussed the development of UserML, an XML-like knowledge representation language developed for the purposes of describing UMs from various application domains. When used as a uniform user modeling language across multiple systems, UserML has a potential to facilitate transfer of UMs in distributed environments and further composition of UMs, accumulated in different systems.

Currently, even though all the required data may be potentially available, systems usually can not assemble comprehensive UMs due to commercial competition, privacy issues, and representation heterogeneity. By using UM mediators that bridge over different approaches and representations, the heterogeneity problem can be solved. Systems will be able to continue using their own methods of UM representation, and yet be able to exchange relevant parts of UMs with other systems, and enrich their UMs.

2 UBIQUITOUS USER MODELING

Personalization systems reside on the Web, in personal devices, and virtually everywhere. Thus, whenever a new user is introduced to a system, it has the potential to gather data about that user from systems all around- this "all-around located" data is ubiquitous in that sense, and the creation of UMs from ubiquitous data is therefore named "ubiquitous user modeling".

A rather simplistic approach of providing personalization in a ubiquitous environment was suggested in [12]. This approach suggested building an application adaptation framework using a personal smart card. The smart card stored and processed a UM, thus partially solving the privacy and availability issues which are essential in decentralized ubiquitous environments. Compared to a solution, where the profiles are stored in a central remote server, the use of smart cards made the profiles available in any context, enhanced privacy and security by allowing the users to fully control their own profiles, and avoided communication delays. However, the smart card remained a single "point of failure", storing sensitive personal information that could be disclosed by a malicious attacker.

Generation of centralized UMs in a ubiquitous environment by composing partial UMs that are stored by different systems was presented by [8]. The paper represented a ubiquitous general UM stored on a central server as a composition of partial UMs, stored by various personalization applications. Every system only maintains the inference mechanism needed for extracting the needed UM data and updating the general model. Although the general model was composed in a distributed manner, it was stored in a central server, which is a single "point of failure" in this case as well.

In [10], the authors highlighted the significance of cross-system personalization that will allow UM data sharing across different systems in a user-centric way. This approach allows information transfer between different systems, and gives the users the ability to control their UMs. Cross-system personalization might be implemented through a central Unified User Context Model (UUCM) [11]. The paper detailed three main stages of a cross-system communication protocol:

- Negotiation – achieving an understanding on the type of information that is needed, i.e., agreeing on common ontology and vocabulary.
- Personalization – extracting data which is relevant to the activity and transferring it to the target system.
- Synchronization – replicating and updating of the stored user model upon completion of personalization tasks.

The involved systems communicated through the mechanism of "context passport" using a mediating architectural layer. UUCM is

[1] The University of Haifa, Haifa, Israel.

based on a shared ontology- each system extracts the required information from the user's passport, performs the required personalization activities, and finally updates the user's passport. To succeed in its mission, the UUCM should have the following two features: generality (to be usable in a variety of domains), and expressiveness (to be able to express a wide variety of facts about the users). The fact that the shared ontology should be developed a-priori is the main drawback of this approach. Furthermore, every system used the ontology, which made it inflexible and unsusceptible to frequent changes.

In [6], the authors introduced GUMO, publicly available General User Model Ontology, which facilitates uniform interpretation of distributed UMs. GUMO is represented through a modern semantic language (OWL [9]) and can be freely used by any personalization system. A common ontology simplifies the exchange of UM data between different systems, which makes it possible to overcome the inherent problem of syntactical and structural heterogeneity between systems. The main problem of GUMO is that it is based on a single central ontology, which prevents dynamicity and frequent changes. Moreover, an initial stage of engineering and construction of a comprehensive all-including ontology requires a vast effort.

To summarize, the above approaches might be insufficient in the dynamic environment of today's information world, since both the available information sources and the needs and interests of the users change frequently, whereas personalization services should keep proper functioning and represent high levels of accuracy. This raises an intriguing research question of developing a mechanism that can easily adapt to the dynamicity of the environment, and at the same time allows the systems to provide an accurate customization of personalized services.

3 USER MODEL MEDIATION

A UM mediator generates UMs on demand, using available users' data according to the specifications of a target system. This is done by querying and receiving partial UMs from various source systems, translating them to the context of the target system, and building an integral UM according to the target system's method of representation from them [1].

Such kind of UM mediator is dynamic in the sense that it is not bound by a specific representation. Any system that requires a UM for bootstrapping may receive it, regardless of the specific personalization technique it uses, and the frequency the UM representation changes, unlike centralized apporaches. Since there are not many techniques used for user modeling, it seems feasible to have a set of specific mediators to be activated in any given scenario.

Another point to be made is that the mediator does not save any data regarding users, thus unlike a personal smart card, user data is not easily breached.

Figure 1 illustrates the stages of the mediation process as described below:

1. A user is requesting a service from a system.
2. In order to provide a personalized service to that user the system requests a UM from the mediator.
3. The mediator identifies the system's application domain and UM representation technique.
4. The mediator extracts from the knowledge base (KB) a set of systems that may provide partial UMs related to the target system's domain.
5. The mediator queries these systems for their UMs of the specific user.
6. Systems that actually store relevant UMs, respond and send the appropriate UM to the mediator.
7. The mediator converts, integrates and assembles the partial UMs (using the KB) into a UM needed by the target application.
8. The generated domain-specific UM is sent to the target system, which is now capable of providing more accurate personalization.

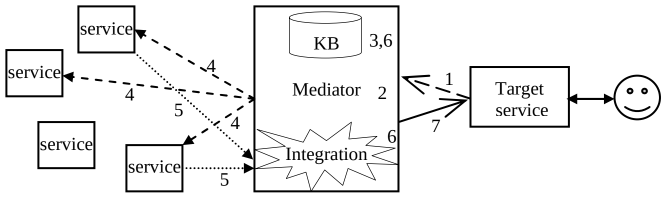

Figure 1. UM Mediator Architecture

4 CURRENT IMPLEMENTATION

A UM mediator system is being developed in the domains of tourism and cultural heritage. This system converts user information from an external trip planning system (Trip@dvice [13]) to a personalized museum visitor's guide system (in use under the PIL project at the Hecht museum at the University of Haifa [16]). In addition, it also serves as an intra-museum UM mediator, since the museum has different exhibitions, whose UM's representation is only partially related.

The UM representation in Trip@dvice is "cased-based". A case is a set of products that a user selected while planning a trip (such as attractions to visit, accommodations etc). In Trip@dvice, the personalization process determines a relevance score for each case item according to the user's current preferences and previous travel plans, and travel plans of users that have similar preferences. In order for a user to select preferred products when planning a trip, descriptions of various offered products, are presented to her/him. The selected products are recorded as a case representing the user's preferences.

Within PIL, each and every exhibit item has several different presentations; each presents the item from a different perspective and the goal of the personalization process is to recommend the user a presentation that best matches his/her interests and information needs. The entire set of terms from the presentations is called "bag of words", thus each presentation is represented as a weighted set of terms. The weights are obtained via the TF*IDF method known from Information retrieval, in which the weight of a term is proportional to the term's frequency in the presentation and to the scarcity of the term in other presentations [14]. A perspective's weighted vector is calculated in a similar way, by considering each perspective as the set of presentations which provide information about various items from that same point of view. The UMs in PIL are "content-based", where a user's preferences are represented by a weighted vector of terms that signify the visitor's preferences, and subsequently by a vector of cosine similarity of the user's UM to the perspectives' representation.

For the purposes of personalization, the presentations that provide information about an item, are being sorted according to their similarity to the visitor's interests (as determined by the UM) before being offered to him/her.

In order to generate a content-based UM from the case-based UM, terms are extracted by the mediator from descriptions of the cases' items in the Trip@dvice user's model. First, the mediator retrieves the cases from the case-based UM. Then, it obtains the free-text case items' descriptions. These descriptions are obtained in two ways: from the KB of tourism attractions, and also seeks out additional information about a case item from the Web. The "bag of words" representation of the UM as extracted from the case is compared to the "bag of words" representation of the exhibition using the well-known cosine similarity metric [14]. This allows deducing features' weights by considering the case item's relevance to the user (as stored in the Trip@dvice system).

Furthermore, every museum's exhibition has its own "bag of words", thus matching terms from one exhibition to another allows intra-museum UM mediation

For the purpose of user modeling, features extracted from acquired case description are converted to the features representing the exhibitions' presentations. First order conversion is aimed at matching the exact same feature (a one to one matching) - the weight of a feature in the target representation is given its weight in the other representation. There are cases in which first order conversion is not good enough, e.g., "ship" and "boat" should probably be considered as the same term even though they are not exactly the same term. Currently, only the first order conversion is implemented in our system.

Second order conversion is aimed at matching between semantically related features. One possibility of doing so is using WordNet [2] to match a feature to a synonym feature. In WordNet, English terms are organized into synonyms sets that describe the same semantic concept. This matching can be enhanced using various machine learning techniques, which weights the relationship between the feature and its synonym features. We are currently working on implementing the second order conversion.

The functional flow of our work is depicted below:

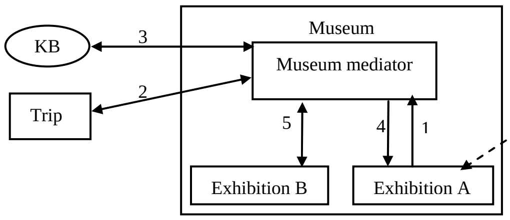

Figure 2. User model mediation in the museum

1. A visitor comes to the museum and enters exhibition *A*. The museum guide system requests an initial UM from the museum mediator.
2. The museum mediator verifies with the trip planning system that the user interacted with the system and retrieves the UM represented by the case items.
3. The mediator addresses an external knowledge-base (KB), and asks for relevant descriptions for the retrieved case items.
4. The mediator assembles a UM by converting features extracted from the case descriptions into UM features in the context of exhibition *A*.
5. When the user arrives to exhibition *B*, the mediator repeats the process and provides with a new conversion of features. The mediator treats the assembled UM of exhibition *A*, as an additional data source.

For example, let us assume that a user's trip planning case consists of the following locations: "Acco's port", "Acco's walls", and the "National maritime museum". These locations have archaeological and historic importance, and the maritime theme is common to them all. Visiting the national maritime museum which has many ship models, might suggest that the visitor is somewhat interested in ship-building methods, thus the technological perspective of the museum presentations, which explains the way that things were made, is probably a good perspective for the user. Let us also assume that the user has started his/her museum visit in the "Phoenicians' exhibition". The descriptions of these user's case locations are collected from the Trip@dvice system. Additional information describing these locations is collected from the Web, or from an external knowledge-base.

Using the Lucene search engine [15] the collected descriptions are indexed and the weight (TF*IDF) is calculated for their terms. The relevance score of a case item to the user is a factor that is considered during the process of determining terms' weights. Some of the collected terms also appear in the presentations of the exhibition, thus in the first order conversion, these terms' weight is conveyed to the generated UM. By using WordNet, every term's synonym set of terms can be found. The exhibition's terms are scanned for the synonym set of terms, which upon detection are added a weight, while taking into account several factors like the size of the term's synonyms set.

During the visit in the "Phoenicians' exhibition" the initial UM is modified to provide better personalization using implicit and explicit feedback from the user. Upon reaching the "Ancient ship exhibition", which has disparate items and consequently dissimilar presentations and only partially related "bag of words" perspectives' representation, the mediator will recalculate a UM relevant to the new exhibition. The mediator will use the UM resulted from the "Phoenicians' exhibition" visit, and generate a new UM for the "ancient ship" exhibition.

5 OPEN QUESTIONS AND FUTURE WORK

Future work will deal with experimentation and evaluation of the mediation in the cultural heritage setting described above. In addition to the obvious benefits of evaluation, the evaluation of the quality of the generated UM will allow the mediator to select between several possible UMs that were deduced from conflicting partial models.

An additional issue to be explored is the decisions the mediator should take regarding the quality of the information available from superficial sources in constructing the requested UM. The confidence level of the mediator in the information sources and in their relevance can be used as a factor while integrating partial UMs (sources that are more trusted will have a larger effect on the generated model). The confidence level can also be provided to the target system allowing it to determine to which extent the provided UM can be trusted and to decide its course of action. In our specific case, the relevance score of a case item might be a starting point to calculate the confidence level of the mediator in the information gathered from that item. The confidence level in the entire generated model might be an accumulation of the confidence levels in the gathered data.

We also intend on addressing third-order conversion, which is aimed at implicitly deducing relationships between the features. For example, if a person is interested in under-water archaeology, he/she might be somewhat interested in diving techniques. We consider using domain ontologies to describe these relationships.

Future work will also extend the current implementation to integrate UMs from several sources, and diverse domains. As stated above, several factors like the confidence level can play a role in the integration of UMs from several sources. For example, if a model of a user already exists in the target system, the mediator might consider it as a very reliable source in the process of generating a more comprehensive model. Other heuristics to solve the question of conflicting partial UMs, e.g., regarding the last time the UM was updated, will be explored.

One might also consider extending the bag of words UM representation in the museum to ontology, as a different form of content-base representation. This could prove to be another interesting case-study.

REFERENCES

[1] S. Berkovsky, T. Kuflik, F. Ricci, "Entertainment Personalization Mechanism through Cross-Domain User Modeling", in proceedings of the INTETAIN Conference, November-December 2005.

[2] C. Fellbaum, “WordNet - An Electronic Lexical Database”, The MIT Press Publishers, 1998.

[3] T. W. Finin, D. Drager, “A general user modeling system”, in Proceedings of the 6th Canadian Conference on Artificial Intelligence, Montreal, Canada, pp. 224-230, 1986.

[4] U. Hanani, B. Shapira, P. Shoval, “Information Filtering: Overview of Issues, Research and Systems”, in User Modeling and User Adapted Interactions, vol. 11 (3), pp. 203-259, 2001.

[5] D. Heckmann, A. Krueger, “A User Modeling Markup Language (UserML) for Ubiquitous Computing”, in Proceedings of the 9th International Conference on User Modeling, Pittsburgh, PA, pp. 393-397, 2003.

[6] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, M. von Wilamowitz-Moellendorff, “GUMO – the General User Model Ontology”, in Proc. of the International Conference on User Modeling, Edinburgh, UK, 2005.

[7] A. Kobsa, W. Pohl, “The BGP-MS user modeling system”, in User Modeling and User-Adapted Interaction, vol. 4(2), pp. 59-106, 1995.

[8] J. Kay, B. Kummerfeld, P. Lauder, “Managing private user models and shared personas”, in Proc. of Workshop on User Modeling for Ubiquitous Computing, Pittsburgh, PA, 2003.

[9] D. L. McGuinness, F. van Harmelen, “OWL Web Ontology Language Overview”, World Wide Web Consortium, 2004. http://www.w3.org/TR/owl-features/

[10] B. Mehta, C. Niederee, A. Stewart, “Towards Cross-system Personalization”, in Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction, Las Vegas, NV, 2005.

[11] C. Niederee, A. Stewart, B. Mehta, M. Hemmje, “A Multi-Dimensional, Unified User Model for Cross-System Personalization”, in Proc. of Workshop on Environments for Personalized Information Access, Gallipoli, Italy, pp 34-54, 2004.

[12] O. Potonniee, “Ubiquitous personalization: A smart card based approach”, in proceedings of the 4th Gemplus Developer Conference, Singapore, 2002.

[13] F. Ricci, B. Arslan, N. Mirzadeh, A. Venturini, “ITR: a Case-Based Travel Advisory System”, in proceedings of the ECCBR Conference, Aberdeen, Scotland, 2002.

[14] G. Salton, M. McGill, Introduction to Modern Information Retrieval. McGraw-Hill, 1983

[15] [Apache Lucene project]http://lucene.apache.org/

[16] [PIL project] http://www.cri.haifa.ac.il/peach.htm.

Web Services and Semantic Web for Adaptive Systems

Francesca Carmagnola, Federica Cena, Cristina Gena, Ilaria Torre
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, Torino, Italy
{carmagnola,cena,cgena,torre}@di.unito.it

Abstract. In this paper we describe the modalities through which an Adaptive System can provide adaptive services using the support of Semantic Web Service technologies. This is the first stage of a project for creating a Semantic Adaptive Web Service that automatically provides both user-adapted services and user models knowledge. In this paper we focus on the enrichment of the service discovery phase through the addition of semantic information.

1 INTRODUCTION

Recent studies have shown a great attention towards standardization and development of interoperable Web technologies. In this direction a considerable number of Web Services (WSs) and Semantic WSs is expected to grow. A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network[1]. A Semantic Web Services enriches every phase of this process of interaction with semantic information. A promising exploitation of Semantic Web Services pertains to i) the possibility of decentralization, communication, integration and interoperability of adaptive applications in order to make possible the exchange of user knowledge [7] and ii) the possibility to define suitable techniques of personalization that support user-centered and preference-based service discovery and selection [2,5,12,13].

The aim of our project is to exploit technologies of Adaptive Systems, Web Services and Semantic Web to develop an *Adaptive Semantic Web Service* which works in a scenario of interoperability with other agents, providing two kinds of output: *i)* personalized services, and *ii)* user model knowledge, exploiting the same knowledge bases and the same technology (Semantic Web Services). In particular, regarding the first kind of output, the system will automatically provide personalized services, allowing software agents to search and to obtain services on behalf of the user (that is particularly important in a mobile environment where the interaction with the device is more challenging). Regarding the second output, the system will automatically provide information concerning its users . This kind of information will be provided to applications that "share" the same users with our application and need some information about them. From this point of view we can say that our application works as a *User Model Web Service*.

Differently with respect to traditional User Modelling Servers, in our approach the user knowledge exchanged across applications is not derived by a server that host user information for different systems, but directly by other adaptive systems, as in Distributed User Model approach [10].

An example of integration of these technologies (adaptive techniques, Semantic Web and Web Service) is provided in the platform developed by [12]. Similarly to this platform, we manage our system exploiting Web Service technology and semantically describing Web Service *inputs* (parameters provided by the agent to make the request possible) and *outputs* (both to *i)* the personalized services and *ii)* the user knowledge the agent may search for). As it will be shown, an essential requirement to achieve these goals is to semantically enrich the description of required inputs and provided outputs.

The project is in initial phase and we started from *output i*, namely the provision of adaptive services [4]. In particular, till now, we focused on the service discovery phase, showing how this phase can be enriched and refined by semantically describing resources and constraints. The objective outlined in this paper is to apply the same approach for the discovery and exchange of user model features (*output ii*).

In detail, the paper will explain how to extend WSDL[2] descriptions with semantic information by using WSSP (Web Service Semantic Profile), a language specifically designed [8] for the description of the service profile and, in our approach, also for the discovery phase of services (*output i*) and for the discovery phase of user models features (*output ii*). Moreover we present the exploitation of OWL[3] and RDF-RuleML[4] to describe ontologies and rules referenced in WSSP. Before this explanation we present a usage scenario and, in the last part of the paper, we provide two examples of WSSP profiles and a RDF-RuleML adaptation rule linked by a WSSP file. These examples come from UbiquiTO-S [4], a Semantic Adaptive Web Service which provides personalised recommendations in the tourist domain, and, as a subsequent step, knowledge about its users (users models knowledge).

2 USAGE SCENARIO

Luca is on holiday in Turin. He visits the city for the first time and is looking for a good restaurant. In order to find out a restaurant he runs the browser of his SmarthPhone GPS-

[1] http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

2 http://www.w3. http://www.w3.org/TR/wsdl
3 http://www.w3.org/TR/owl-features/
4 http://www.w3.org/2004/12/rules-ws/paper/93/

equipped and queries his favorite matchmaker[5]. From the search interface of the matchmaker, he selects the categories tourism and restaurants. The matchmaker already knows some information about Luca, such as his position (provided by his GPS-receiver) and his employment and age (provided by Luca when he registered to the matchmaker's search service). Several Web Services provide tourist information. Among the others, the matchmaker discovers UbiquiTO-S, a Semantic Adaptive Web Service that provides personalized touristic recommendations regarding restaurants. Thus, without providing any other information nor about himself nor about his goals, thanks to the interaction between UbiquiTO-S and the matchmaker, Luca receives a list of restaurants that mostly correspond to his preferences. He is satisfied because the list of the restaurants fits his needs, and also because he did not spend too much time browsing web sites about restaurants in Turin, since the matchmaker have automatically carried out this search on his behalf.

3 SEMANTIC DESCRIPTION OF PARAMETERS

The above example presents a scenario where several Web Services (UbiquiTO-S is one of them) provide services to end-users through software agents. As typical, they describe their services in WSDL files and advertise them in public UDDI[6] registries. An agent (the matchmaker in the scenario) searches the UDDI registries looking for Web Services able to satisfy Luca's request. The matchmaker evaluates both the correspondence between the user requests (recommended restaurants for Luca, in the scenario) and the systems outputs supplied by Web Services, and between Web Service requirements and the available user features (Luca's age and employment). Then, it decides to invoke UbiquiTO-S services since UbiquiTO-S is able to provide not only location based information, as done by the other discovered Web Services, but also information tailored to the user's need (similarly, considering *output ii*, a software agent which needs user features can search for Web Services which provide such features for a specific user and decide to invoke the service which better satisfies its needs). On the basis of the available information (i.e., age and employment), UbiquiTO-S makes inferences about some other features (i.e., the user salary), and finally fires its adaptation rules to extract and provide personalized suggestions to the user.

In order to obtain that, Ubiquito-S is implemented as a Web Service that semantically annotates provided outputs, required inputs and their relationships. In particular, UbiquiTO-S enriches the WSDL service description by means of a WSSP file. WSSP has been introduced and successfully exploited by [8]. It is a discovery mechanism that semantically enriches the UDDI's search functionality and encodes semantic information with WSDL. WSSP is able to specify ***restrictions*** and ***constraints*** for WSDL parameters of input and output by describing the service capabilities with semantic information expressed in some ontology language. Many other standards have been proposed (e.g., OWL-S[7], WSDL-S[8], WSML[9], SWSL[10]), but we chose WSSP since it is compatible with the service ontology of OWL-S, it allows to use semantic rule languages (like RDF-RuleML) also in the description of the service profile, and it can be complementary used with the standard WSDL and UDDI structures.

To illustrate how this technology can be applied to adaptive systems, the following part of the paper will describe input and output messages, specifying:

a) how they are defined in common WSDL files,

b) how WSSP allows to semantically specify them.

For each one we will explain how this specification can be instantiated and fruitfully exploited to obtain **output i** and **output ii**.

INPUT

a) According with Web Service standards, the input message of WSDL files specifies parameters, typically domain/service features, used to specify the requested service. Considering *Adaptive Web Services*, they could require, as input, not only parameters regarding the domain, but also parameters regarding the user profile (e.g. age, goals, etc.), and the current user context (e.g. position, direction, state, etc.). Considering a *User Model Service*, input message specify the required user features or even the required context user features (for an example of exchange of context user features, my means of Web Services see [11]).

In the scenario previously described, input parameters specify the kind of service the user searches for, but also user features such as the user position, age and employment.

b) WSSP file is exploited to specify restrictions and constraints to the above-mentioned WSDL parameters.

Restrictions concerning input allow specifying the WSDL parameters by mapping them on an ontology. This is particular relevant in a context of interoperability, since an agent has to exactly know the required input in order to compare it with the data it owns and to decide if it is able to invoke the Web Service.

The following code is a part of the WSSP file which specifies (and thus restricts) the input parameter *employment* by referring it to the identifier *Employment* of the shared and public ontology *UserModelOntology.owl* [6] (thus specifying its meaning, super-classes, relations, etc.).

Note that the same ontology could be used to specify the required user features to a *User Model Web Service*.

WSSP description (example of restriction)
```
<profile:parameterName>Employment
</profile:parameterName>
```

5 A matchmaker is a search engine a user can delegate to find services. It works performing a match between the user request and the service description typically advertised on UDDI registries.

6 http://www.uddi.org/

7 http://www. daml.org/services/owl-s/

8 http://www.w3.org/2005/04/FSWS/Submissions/17/WSDL-S.htm

9 http://www.wsmo.org/wsml/

10 http://www.daml.org/services/swsl/

```
<profile:restrictedTo
rdf:resource="http://u2m.org/2003/02/UserModelOnto
logy.owl#Employment"/>
```

Constraints to input message may be expressed as *Facts* or *Rules*. They can involve one or a set of parameters and can be described by means of semantic rule languages such as RuleML. Considering their use in *Adaptive Web Service*, they can be useful to better specify the data and allowed values that will be used for the user modeling task. For example, they may specify the source of information, the lowest threshold of confidence for the provided input, etc. The advantage is the possibility to improve the quality of the provided input since in this way the required inputs are better specified. Considering *User Model Web Services*, this possibility is extremely useful since it allows, for example, to face privacy issues, specifying access restrictions and authentication requirements.

OUTPUT

a) Besides input parameters, WSDL files specify output parameters, which concern the service provided by the Web Service (for **output i**). Considering **output ii**, that is services provided by a User Model Service, WSDL files will specify the user models features the Web Service provides.

b) As for inputs, also for outputs the WSSP structure allows specifying restrictions and constrains.

Restrictions allow specifying the parameters by mapping them on an ontology. This specification allows improving the service search, exploiting also information of generalization/specification provided by the ontology.

Constraints allow specifying conditions on the provision of the service. They can be used, for example, to specify the opening days and times, delivery days conditioned to the kind of required service or product, etc.

Considering *Adaptive Semantic Web Services*, constraints could also be usefully exploited to specify rules that explain the reason why inputs are required in order to obtain outputs and how output changes on the basis of the provided inputs. For example some rules exploited in the above scenario specify that:

- if the user employment is provided (input), suggestions regarding events (output) have a Confidence Value (CV) of 0.5;

- if the user employment and age are provided (input), suggestions regarding events (output) have a CV of 0.7;

Considering *User Model Web Services*, the possibility to specify constraints is very important, since it could allow the requestor to know the relevance of the feature according to the goal of the request.

Below we present a part of a WSSP file where it is shown the use of the tag constraint and its reference to an URI which specifies a RDF-RuleML rule. As it can be seen, this rule expresses in RDF-RuleML language the second rule of the above examples.

WSSP description (example of constraint)

```
<profile:output>
  <profile:message rdf:ID="ServiceResponse">
   <profile:constrainedBy
rdf:resource="http://ubiquito-s
/rules.rdf#ProvideAgeEmployment"/>
  </profile:message>
</profile:output>
```

RDF-RuleMl Rule

```
<Implies>
 <rdf:Description ref:about="http://ubiquito-s
/rules.rdf#ProvideAgeEmployment">
 <rdf:type resource="http://ubiquito-
s/preds/Relation.owl#ProvideAgeEmployment">
 <rdf:type resource="http://ubiquito-
s/preds/Relation.owl#HasConfidence">
 </rdf:Description>
<body> <Atom>
    <oid><Ind wlab=http://ubiquito-
s/UserModel.owl#User /></oid>
    <opr><Rel wref="http://ubiquito-
s/preds/Relation.owl#ProvideAgeEmployment"></opr>
    <slot><Ind wref="
http://u2m.org/2003/02/UserModelOntology.owl#Age"/
></slot>
    <slot><Ind wref="
http://u2m.org/2003/02/UserModelOntology.owl#Emplo
yment"/></slot>
 </Atom> </body>
<head>  <Atom>
    <oid><Ind wlab="http://ubiquito-s"/></oid>
    <opr><Rel wref="http://ubiquito-
s/preds/Relation.owl#HasConfidence"></opr>
    <slot><Ind wref=" http://ubiquito-
s/onto/Output.owl#Events"/></slot>
    <slot><Ind
wref="http://www.u2m.org/2003/situation#confidence
"/><Data>0.7</Data></slot> </Atom></head>
</Implies>
```

4 CONCLUSION AND FUTURE WORK

In this paper we presented UbiquiTO-S, an adaptive system which exploits the technology of Web Services and Semantic Web to allow software agents to discover and invoke its adaptive services. UbiquiTO-S has been developed as an extension of UbiquiTO [1], a mobile adaptive guide that provides personalized location-based tourist information. With respect to UbiquiTO, the core functionalities are not changed. However, its transformation into a Web Service with a formal and semantic description of input requirements, outputs and their relationships introduces it into a cooperative environment for personalized services.

With respect to common Web Services, the advantage of this approach is that it allows, for instance, a middle agent (the matchmaker in the scenario) to obtain services that fit user features and can be imported and processed to compose other services which take advantage of the personalization. Moreover, considering the user point of view, while UbiquiTO provides adapted information without explaining the reason why it needs some data, UbiquiTO-S, on the contrary, is more scrutable since it allows to know how outputs depend on inputs.

As specified in the introduction, this work is the first stage of a project aimed to exploit the technology of Web Services and Semantic Web in order to build an Adaptive Web Service which provides two kinds of outputs: adaptive services and user model features. This initial phase of our work has been focused on the first type of output and in particular on the semantic discovery phase of the service (*output i*), by adding semantic information to the service profile. The current work includes the definition of the WSDL file, of the WSSP profile, of the OWL ontologies and of the RDF-RuleML rules that are referred in WSSP to specify restrictions and constraints of the input and output messages. Our next step will deal with adding semantics to the whole process that includes service composition, execution and monitoring. Then, we will apply the same approach to offer not only personalized service, but also knowledge concerning the user (*output ii*) taking into account the user privacy management [9]. The aim is, to move towards the creation of a *User Models Service*, a Web Service that provides information about the inferred user features (and the reasoning strategies that lead to infer user features, as explained in [3]).

REFERENCES

1. Amendola, I., Cena F., Console L., Crevola A., Gena C., Goy A., Modeo S., Perrero M., Torre I., Toso A.: UbiquiTO: A Multi-device Adaptive Guide. Proc. of Mobile HCI 2004, LNCS, 3160 (2004) 409-414.
2. Balke, W., Wagner, M.: Towards personalized Selection of Web Services. In Proc. World Wide Web Conf. (WWW), Budapest, 2003.
3 Carmagnola, F., Cena, F., From Interoperable User Models to Interoperable User Modeling. To appear in Proc. of AH 06, Ireland, June 06.
4. Carmagnola, F., Cena, F., Gena, C., Torre, I.: UbiquiTo-S: a preliminary step toward Semantic Adaptive Web Services. To appear in the Proc. of AH 06, Ireland, June 2006.
5. De Bra, P., Aroyo, L., Chepegin, V.: The Next Big Thing: Adaptive Web-Based Systems. In Journal of Digital Information, Vol. 5 Issue 1 N.247, 2004-05-27.
6. Heckmann, D., Ubiquitous user modeling for situated interaction. In Proc. International Conference on User Modeling, Berlin Heidelberg, 2001.
7. Heckmann, D. Ubiquitous User Modeling. PhD thesis, Department of Computer Science, Saarland University, Germany, 2005
8. Kawamura, K., De Blasio, J.-A., Hasegawa, T., Paolucci, M., Sycara, K., Preliminary Report of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry, Lecture Notes in Computer Science, Volume 2910, Jan 2003, p 208– 224.
9. Kay, J., Kummerfeld, B., and Lauder, P., Managing private user models and shared personas. In UM03, Pittsburgh, USA (2003).
10. Lorenz, A., A Specification for Agent-Based Distributed User Modelling in Ubiquitous Computing, in Proc. DASUM-05, Edinburgh, UK (2005).
11. Luther, M., Boehm, S., Wagner, M. and Koolwaaij, J., Enhanced Presence Tracking for Mobile Applications, In Proc. of ISWC'05 Demo Track, Galway (2005).
12. Van Setten, M., Pokraev, S., Koolwaaij, J.: Context-aware recommendations in the mobile tourist application COMPASS. In Nejdl, W. & De Bra, P. (Eds.). AH 2004, 26-29 Aug 04, Eindhoven, The Netherlands, LNCS 3137, Springer-Verlag, pp. 235-244.
13. Wagner, M., Balke, W., Hirschfeld, R., Kellerer, W.: A Roadmap to Advanced Personalization of Mobile Services. In Industrial Program of CoopIS 2002, Irvine, USA, 2002.

Efficient Text Summarization for Web Browsing On Mobile Devices

Gaël Dias[1] and Bruno Conde[2]

Abstract. In this paper, we propose an automatic summarization server-based architecture for web browsing on handheld devices. In particular, we introduce different efficient methods for summarizing parts of web pages in real-time. Two main approaches have already been proposed in the literature. First, some methodologies such as [1] [5] use simple summarization techniques to produce results in real-time but clearly lack linguistic treatment for reliable content visualization. Second, some works apply linguistic processing and rely on *ad hoc* heuristics [2] to produce compressed contents but can not be used in real-time environment. As a consequence, we propose a new architecture for summarizing Semantic Textual Units [1] based on efficient algorithms for linguistic treatment that allow real-time processing and deeper linguistic analysis of web pages, thus allowing quality content visualization.

1 INTRODUCTION

The shift in human-computer interaction from desktop computing to mobile real-world interaction highly influences the needs for future decentralized user-adaptive systems. Designing personalized Web Services such as text summarization for web browsing on mobile devices is one of many challenges for the success of ubiquitous computing.

For handheld devices, screen size limitation is clearly the issue as most web pages are designed to be viewed on desktop displays. Indeed, the smallest web page excerpts displayed on any mobile device screen can interfere with users' comprehension, and the resulting scrolling is time consuming.
Some solutions have been proposed to overcome these limitations. They usually require an alternate trimmed-down version of documents prepared beforehand (e.g. WAP Browsers) or the definition of specific formatting styles (e.g. XML Schemas). However, this situation is undesirable as it involves an increased effort in creating and maintaining alternate versions of a web site.

To solve this problem, we propose an automatic summarization server-based architecture for web browsing on handheld devices. In particular, we introduce four different efficient methods for summarizing subparts of web pages in real-time. Two main approaches have already been proposed in the literature. First, some methodologies such as [1] [5] use simple but fast summarization techniques to produce results in real-time. However, they show low quality contents for visualization as they do not linguistically process the web pages. Second, some works apply linguistic processing and rely on *ad hoc* heuristics [2] to produce compressed contents but can not be used in a real-time environment. Moreover, they do not use statistical evidence which is a key factor for high quality summarization. As a consequence, we propose a new architecture, called XSMobile, for summarizing Semantic Textual Units [1] based on efficient algorithms for linguistic treatment [3] [4] that allow real-time processing and deeper linguistic analysis of web pages, thus producing quality content visualization as illustrated in Figure 1.

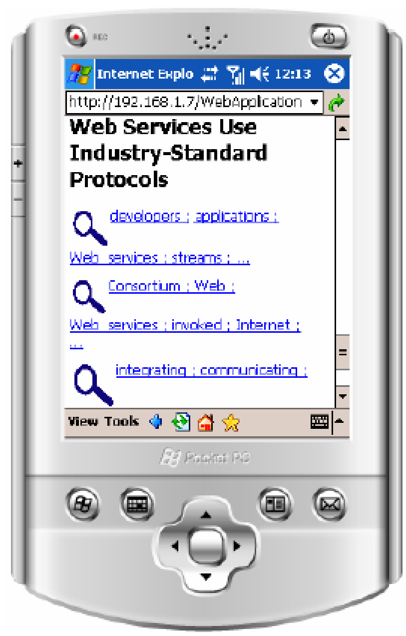

Figure 1. Screenshot of the XSMobile architecture

This paper is divided into five sections. First, we present the relevant related work in the area. Second, we talk about text unit identification and review the concept of Semantic Textual Units proposed by [1]. Third, we emphasize the linguistic treatment we apply on each Semantic Textual Units. Fourth, we present some implemented summarization techniques. And finally, we explain how the information is displayed on the mobile device.

2 RELATED WORK

[1] is certainly the most relevant first appearing paper of this field. They introduced two methods for summarizing parts of web pages. Each web page is broken into Semantic Textual Units that can each be hidden, partially displayed, made fully visible, or summarized. However, their work is built on old well known techniques for text summarization and do not introduce linguistic processing (except stemming) to remain real-time adaptable as processing is handled by the mobile device.

In order to introduce more knowledge compared to the previous model, [5] propose a fractal summarization model based on statistical and structure analysis of web pages. Thus, thematic features, location features, heading features, and cue features are

[1] Centre of Human Language Technology and Bioinformatics, University of Beira Interior, Portugal, email: ddg@di.ubi.pt
[2] Centre of Human Language Technology and Bioinformatics, University of Beira Interior, Portugal, email: countbruno@gmail.com

adopted. Their architecture first generates a skeleton of a summary and its details are generated on demands of users. Comparatively to [1], [5] propose a more organised structure but do not use any linguistic processing although they work on basis of a three-tier architecture which provides more processing power.

[2] are the first to introduce some linguistic knowledge into the process of text summarization. They use a parser to perform text segmentation and morphological analysis. In particular, they apply linguistic patterns for sentence compression rather than for sentence extraction. For example, some names are replaced with their acronyms and some adjectives may also be removed. The major drawback of this approach is the lack of statistical analysis which is a key factor for high quality summarization.

In XSMobile, our objective is to use both statistical evidence and linguistic processing for sentence extraction in real-time. For that purpose, we use two efficient linguistic softwares (the TnT tagger [3] and the SENTA multiword unit extractor [4]) and propose new sentence weighting schemes. To our knowledge, this is the first attempt to use both statistical and linguistic techniques for text summarization for browsing on mobile devices.

3 TEXT UNITS IDENTIFICATION

One main problem to tackle is to define what to consider as a relevant text in a web page. Indeed, the summary of a web page will be created on the basis of the text extracted by the web server. However, web pages often do not contain a coherent narrative structure [7]. So, the first step of any system is to identify rules for determining which text should be considered for summarization and which should be discarded.

For that purpose, [8] propose a C5.0 classifier to differentiate narrative paragraphs from non narrative ones. However, 34 features need to be calculated for each paragraph which turns this solution impractical for real-time applications.

In the context of automatic construction of corpora from the web, [9] propose to use a language model based on Hidden Markov Models (HMM) using the SRILM toolkit [10]. This technique is certainly the most reliable one as it is based on the essence of the language but still needs to be tested in terms of processing time[3].

Finally, [1] propose Semantic Textual Unit (STU) identification. In summary, STUs are page fragments marked with HTML markups which specifically identify pieces of text following the W3 consortium specifications. However, not all web pages respect the specifications and as a consequence text material may be lost. In this case, unmarked strings are considered STUs if they contain at least two sentences. It is clear that the STU methodology is not as reliable as any language model for content detection but on the opposite it allows fast processing of web pages.

So, any requested web page is first divided into STUs (i.e. narrative paragraphs) so that further linguistic processing can be performed to identify relevant information about the text.

4 LINGUISTIC PROCESSING

On the one hand, single nouns and single verbs usually convey most of the information in written texts. They are the main contributors to the "aboutness" of any text. On the other hand, compound nouns (e.g. *hot dog*) and phrasal verbs (e.g. *take off*) are also frequently used in everyday language, usually to precisely express ideas and concepts that cannot be compressed into a single word. So, compound nouns and phrasal verbs provide good clues for text content description. As a consequence, identifying these lexical items is likely to contribute to the performance of the extractive summarization process [11]. For that purpose, we apply to each STU the following linguistic treatment.

Each STU in the web page is first morpho-syntactically tagged with the TnT tagger [3] which is an implementation of the Viterbi algorithm for second order Markov Models [12]. The main paradigm used for smoothing is linear interpolation and respective weights are determined by deleted interpolation. Unknown words are handled by a suffix trie and successive abstractions. As a summary, TnT is an efficient tagger in terms of processing power and reaches precision results around 96% to 99%.

Once morpho-syntactically tagged, each STU is processed by the SENTA multiword unit extractor [4]. SENTA combines an association measure called Mutual Expectation with an acquisition process based on an algorithm of local maxima called GenLocalMaxs over a data set of positional ngrams. Its efficient implementation shows time complexity $\Theta(N \log N)$ where N is the number of words to process. It is based on the definition of masks that virtually represent any positional ngram in the text and applies a suffix-array data structure coupled with the Multikey Quicksort algorithm [13] to compute positional ngram frequencies in real-time.

Both softwares are freely available and flexible for any language as the TnT can be trained on any tag set and SENTA is an unsupervised statistical parameter-free architecture. This is an important remark as our architecture can easily be adapted to other languages and as a consequence is totally portable.

Then, we apply some heuristics to define quality multiword units for content visualization. So, multiword units that do not respect the following regular expression are filtered out:

[Noun Noun* | Adjective Noun* | Noun Preposition Noun | Verb Adverb].

This technique is usual in the field of Terminology [14]. A good example can be seen in Figure 1 where the multiword unit "Web Services" is detected, where existing solutions would at most consider both words "Web" and "Services" separately. This would lead to less expressiveness of the content of the STU and may imply text understanding errors.

Finally, we remove all stop words present in the STU. This process allows faster processing of the summarizing techniques as the Zipf's Law [15] shows that stop words represent 1% of all the words in texts but cover 50% of its surface.

5 SUMMARIZATION TECHNIQUES

Once all STUs have been linguistically processed, the next step of the extractive summarization architecture is to extract the most important sentences of each STU. In order to make this selection, each sentence in a STU is assigned a significance weight. The sentences with higher significance become the summary candidate sentences. Then, the compression rate chosen by the user defines the number of sentences to present on the screen of the device.

[3] By the time of implementation, this solution was unknown to us and as a consequence was not considered, but will be tested in future work.

For that purpose, we implement four basic extractive techniques: the simple tf.idf, the enhanced tf.idf and the two methodologies proposed by [1]. It is clear that more powerful methodologies exist. However, there are not still tailored for fast processing [11], although some research is done in this direction [16].

In the following subsections, we will explain the simple tf.idf and the enhanced tf.idf methodologies and introduce the cluster methodology proposed by [1].

5.1 Simple tf.idf

This methodology is simple and mainly used in Information Retrieval [6]. The sentence significance weight is the sum of the weights of its constituents divided by the length of the sentence.

A well-known measure for assigning weights to words is the tf.idf score [17]. The idea of the tf.idf score is to evaluate the importance of a word within a document based on its frequency and its distribution across a collection of documents. The tf.idf score is defined in Equation 1 where *w* is a word, *stu* a STU, *tf(w, stu)* the number of occurrences of *w* in *stu*, |*stu*| the number of words in the *stu* and *df(w)* the number of documents where *w* occurs.

$$tf.idf(w, stu) = \frac{tf(w, stu)}{|stu|} \times \log_2 \frac{N}{df(w)} \tag{1}$$

In our case, we processed all idf[4] values from a collection of texts: the DUC 2004 collection[5] plus all the texts in our test website. In particular, all texts of the collection have been linguistically processed as explained in Section 4.

So, the sentence significance weight, $weight_1(S, stu)$, is defined straightforwardly in Equation 2

$$weigth_1(S, stu) = \frac{\sum_{i=1}^{|S|} tf.idf(w_i, stu)}{|S|} \tag{2}$$

where $|S|$ stands for the number of words in S and w_i is a word in S.

5.2 Enhanced tf.idf

In the field of Relevant Feedback, [6] propose a new score for sentence weighting that proves to perform better than the simple tf.idf. In particular, they propose a new weighting formula for word relevance, *W(.,.)*. In fact, this is a refinement of the tf.idf measure and it is defined in Equation 3

$$W(w, stu) = \left(0.5 + \left(0.5 \times \frac{tf(w, stu)}{\arg\max_{w}(tf(w, stu))}\right)\right) \times \log_2 \frac{N}{df(w)} \tag{3}$$

where argmax(*tf*(*w,stu*)) corresponds to the word with the highest frequency in the STU.

Based on this weighting factor, [9] define a new sentence significance factor $weight_2(S, stu)$ that takes into account the normalization of the sentence length. The subjacent idea is to give more weight to sentences which are more content-bearing and central to the topic of the STU i.e. which contain a higher proportion of words with high tf.idf as shown in Equation 4

$$weigth_2(S, stu) = \frac{\sum_{i=1}^{|S|} W(w_i, stu)}{\left(\dfrac{\arg\max_{s}(|S|)}{|S|}\right)} \tag{4}$$

where argmax(|*S*|) is the length of the longest sentence in the STU.

5.3 Cluster methodologies

Luhn suggested in [19] that sentences in which the greatest number of frequently occurring distinct words are found in greatest physical proximity to each other, are likely to be important in describing the content of the document in which they occur. [1] based their sentence ranking module on this paradigm.

The procedure proposed by [1], when applied to sentence *S*, works as follows. First, they mark all the significant words in *S*. A word is significant if its tf.idf is higher than a certain threshold *T*. Second, they find all clusters in *S* such that a cluster is a sequence of consecutive words in the sentence for which the following is true: (i) the sequence starts and ends with a significant word and (ii) fewer than *D* insignificant words must separate any two neighboring significant words within the sequence. This is illustrated in Figure 2 where "*" are significant words and *D=2*.

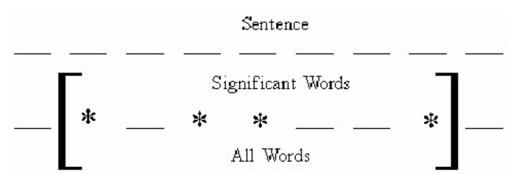

Figure 2. Cluster representation taken from [1].

Then, a weight is assigned to each cluster. This weight is the sum of the weights of all significant words within a cluster divided by the total number of words within the cluster. Finally, as a sentence may have multiple clusters, the maximum weight of its clusters is taken as the sentence weight.

6 VISUALIZATION

The last part of the process is the visualization phase. For that purpose, the user can choose one option from a set of five levels of visualization for each summarization methodology as shown in Figure 3. In particular, at installation time, a link to this configuration page is automatically inserted in each page of the website. As a consequence, the user can choose a different visualization mode for each browsed web page. This mechanism is handled by cookies.

Following the same strategy as in [1], the user can choose between the following five options: (1) first characters of the most relevant sentence in the STU[6] and no summarization, (2) five most relevant keywords[7] in the STU and no summarization, (3) first characters of the most relevant sentence in the STU and summarization, (4) five

[4] The idf is the second argument of the product in Equation 1.
[5] The DUC 2004 corpus is available at http://duc.nist.gov/.

[6] This is the same idea as web snippets.
[7] Here, keyword stands for the most relevant lexical items in the STU according to the word weighting factor.

most relevant keywords in the STU and summarization, (5) no processing of the web page.

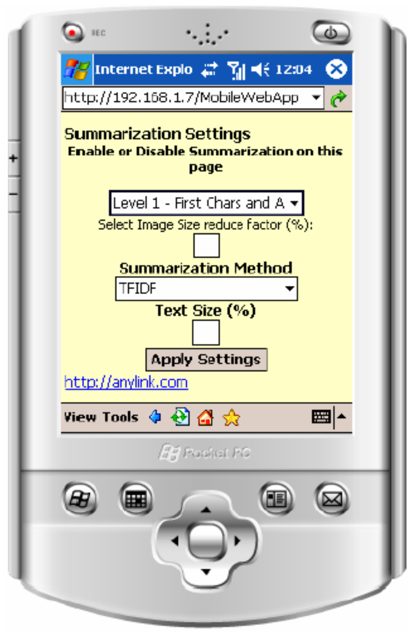

Figure 3. Screenshot of the XSMobile configuration page.

It is clear that for levels 3 and 4, the user must define the summarization compression rate, *C*. Each STU is then summarized according to *C* i.e. $E(C.|S|)$ significant sentences are presented in order of relevance where $E(.)$ is the floor function, *C*, the compression rate and |S| the number of sentences in the STU.

In order to help the user in its search for information, we also define a degree of significance of each STU. So the more relevant a STU is, the bigger its associated magnifying glass will be as shown in Figure 1. The significance factor of a STU is simply calculated as in Equation 5

$$factor_j(stu) = \sum_{i=1}^{|S|} weight_j(S_i, stu) \qquad (5)$$

where j ($j=1..4$) defines the significance sentence weight formula. This weight is then normalized among all STUs in the web page so that its value ranges between [0..1] i.e. it represents its percentage of relevance compared to all other STUs relevance weights.

Finally, image compression rate is also accessible to the user. In this case, the process is performed by reformulating the <img> tag i.e. by modifying/inserting the width and height attributes. This process reduces both the size of the picture on the screen and the size of the picture to be transferred on the network. We are aware that this compression rate is not ideal but some improvements will be introduced in future work.

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed an automatic summarization server-based architecture for web browsing on handheld devices. Unlike previous works [1] [2] [5], it is based on efficient algorithms [3] [4] for linguistic treatment that allow real-time processing and deeper linguistic analysis for quality content visualization. The first results are every encouraging in terms of (1) quality of the content of the summaries, especially with the enhanced tf.idf, (2) processing time although the architecture is not still distributed over different processing units and (3) user interaction satisfaction. However, many improvements must be taken into account. Immediate future work involves applying a language model for content detection instead of the STU strategy. Another important improvement has to do with document structure. Indeed, hierarchical display is suitable for navigation of large documents and it is ideal for small area displays [5]. But, unlike [5], we intend to apply a hierarchical graph-based overlapping clustering algorithm [18] to automatically infer from text content only the relationships between text subparts.

ACKNOWLEDGMENTS

This work is funded by the Portuguese Foundation for Science and Technology under the SUMO project - POSC/PLP/57438/2004.

REFERENCES

[1] O. Buyukkokten, H. Garcia-Molina and A. Paepcke. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In Proc. of the 10th International World Wide Web Conference. (2000).
[2] P. Gomes, S. Tostão, D. Gonçalves and J. Jorge. Web-Clipping: Compression Heuristics for Displaying Text on a PDA. In Proc. of 3rd Workshop on Human Computer Interaction with Mobile Devices. (2001).
[3] T. Brants. TnT - a Statistical Part-of-Speech Tagger. In Proc. of the 6th Applied NLP Conference. (2000).
[4] A. Gil and G. Dias. Using Masks, Suffix Array-based Data Structures and Multidimensional Arrays to Compute Positional Ngram Statistics from Corpora. In Proc. of the Workshop on Multiword Expressions of the 41st ACL. (2003).
[5] C. Yang and F.L. Wang. Fractal Summarization for Mobile Devices to Access Large Documents on the Web. In Proc. of the International World Wide Web Conference. (2003).
[6] O. Vechtomova and M. Karamuftuoglu. Comparison of Two Interactive Search Refinement Techniques. In Proc. of HLT-NAACL. (2004).
[7] A. Berger and V. Mittal. Ocelot: a System for Summarizing Web Pages. In Proc. of SIGIR. (2000).
[8] Y. Zhang, N. Zincir-Heywood, and E. Milios. Summarizing Web Sites Automatically. In Proc. of the 16th Conference of the Canadian Society for Computational Studies of Intelligence. (2003).
[9] W. B. Dolan, C. Quirk and C. Brockett. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In Proc. of COLING. (2004).
[10] A. Stolcke. SRILM -- An Extensible Language Modelling Toolkit. In Proc. of International Conference on Spoken Language Processing. (2002).
[11] R. Barzilay and M. Elhadad. Using Lexical Chains for Text Summarization. In Proc. of the Intelligent Scalable Text Summarization Workshop of ACL. (1997).
[12] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In IEEE. (77)2. (1989).
[13] J. Bentley and R. Sedgewick. Fast Algorithms for Sorting and Searching Strings. In Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms. (1997).
[14] J. Justeson and S. Katz. Technical Terminology: some Linguistic Properties and an Algorithm for Identification in text. Natural Language Engineering, (1). (1995).
[15] G. K. Zipf. Selective Studies and the Principle of Relative Frequency in Language. Harvard University Press. (1932).
[16] G. Silber and K. McCoy. Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. Computational Linguistics. (28)4. (2002).
[17] G. Salton, C.S. Yang, and C.T. Yu. A Theory of Term Importance in Automatic Text Analysis. Amer. Soc. of Inf. Science. (26)1. (1975).
[18] G. Cleuziou, L. Martin and C. Vrain. PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classication and Textual Data. In Proc. of the 16th EACL. (2004).
[19] H.P. Luhn. The automatic creation of literature abstracts. IBM Journal of Research and Development.(1958).

Creating Ontology for User Modelling Research

Tatiana Gavrilova[1], Peter Brusilovsky[2], Michael Yudelson[2] and Seppo Puuronen[3]

Abstract. The paper proposes meta-ontology of the user modelling field. This structured approach based on methodology of knowledge engineering may facilitate the research activity. It features ontology or conceptual domain structure design and refinement as a kernel and guide for a fruitful and meaningful research roadmap. Ontology is meant to structure the state-of-the-art in the field and serves as a central reference point and as guiding tool to do research, index systems, papers and learning media. Such ontology may help graduate and post-graduate students to shape the proper framework for their study of such multi-faceted and multi-disciplinary field as "User Modelling".

1 INTRODUCTION

Doing research is a creative and challenging activity. Young researchers use a lot of informal rules-of-thumb advice that may help but not a systematic guidelines. This paper presents a framework that may be helpful for the self-guided students and young supervisors. In addition, it details an approach to promoting integrity during the education of researchers, including how to develop an effective thesis. Providing a framework for research in user modelling (UM) this paper may be essential for anyone concerned about methodology of scientific study.

All researchers are knowledge workers. Ontology may serve as a common language to all of the researchers as ontology is an explicit specification of a conceptualization [11]. Describing ontology is describing the skeleton of the problem. It is not only a description, it is an explanation and clarification of new knowledge that was worked out. From a philosophical viewpoint, "ontology" (without the indeterminate article and with the uppercase initial) is the branch of philosophy which deals with the nature and the organization of reality.

The area of user modelling (UM) currently possesses a great deal of heterogeneity. The terminology is still not standardized. A lot of terms have multiple synonyms (e.g. behavioural user model, feature-based user model, individual user model) and terms are often fuzzy.

There exist a lot of approaches to UM but a common schema that would attempt to classify them all has not been proposed yet. Such lack of structure makes attempts to conduct novel research or implement known approaches in the area of UM quite a demanding task.

This is why we propose a classification, an ontology of UM field that may work as a central reference point, just like ACM computing classification system (ACM CCS) [2]. Such ontology helps to present the current state of the art in a visual structured form and may serve as a teaching tool or the basis for comparing papers and pieces of research in the specific area of UM. Now ontologies aim at capturing domain knowledge in a generic way and providing a commonly agreed understanding of a domain, which may be reused and shared across applications and groups [10].

2 ONTOLOGICAL ENGINEERING – FIRST STEPS.

Neches and colleagues [19] gave classical definition as follows "An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary". Fig.1 illustrates our view on main ontological classifications.

Ontology is a set of distinctions we make in understanding and viewing the world. Visual approach to present ontology is not only compact but also very comprehensive. It makes ontology a powerful mind tool [15, 10].

Using some of design criteria set of principles that have been proved as useful in the development of ontologies [11, 17, 21] and generalising our experience in ontology development [10] we can propose one simple recipe in a nutshell comprising 4 steps for visual ontology development:

1. *Glossary development* – selecting and verbalizing the essential domain concepts.
2. *Laddering* – defining the main levels of abstraction. It is also important for the next stages of the design to elucidate the type of ontology according to classification.
3. *Disintegration/Categorization* –breaking high level concepts into a set of detailed when needed (top-down strategy) and associating similar concepts to generalize meta-concepts (bottom-up strategy.
4. *Refinement* – updating the visual structure by excluding the excessiveness, synonymy, and contradictions.

The main goal of the above mentioned algorithm is to create a visually appealing ontology that means that ontology developer should observe conceptual balance ('harmony') and clarity of the ontology [6,10]. A well-balanced ontological hierarchy equals a strong and comprehensible representation of the domain knowledge. Here are our tips on how to achieve 'harmony'. First, sibling concepts should present objects of the same level of granularity and should be linked to the parent concept by one type of relationship. Second, the depth of the branches should be more or less equal (±2 levels). Third, the general outlay should be symmetrical. Fourth, cross-links should be avoided as much as possible.

Ontology clarity can be achieved by optimizing the number of concepts and types of the links between them. Minimizing the number of concepts is the best tip according to Ockham's razor principle. The maximal number of branches and the number of levels should also follow well-known 7±2 rule by Miller. The type of relationship should be clear and obvious if the name of the relationship is omitted.

Such balanced ontological framework scaffolds the research activity.

[1] St.Petersburg State Polytechnic University, tgavrilova@gmail.com
[2] University of Pittsburgh, {yudelson, peterb}@mail.sis.pitt.edu
[3] University of Jyväskylä, Finland, sepi@cs.jyu.fi

3 ONTOLOGIES FOR RESEARCH

Traditional approach to research may be formulated in general as a set of 8 sequential steps proposed by Jenkins [14, 15] to describe major research processes. It is a simplification as the process is really iterative:

1. Idea
2. Library research
3. Research topic
4. Research strategy
5. Experimental design
6. Data capture
7. Data analysis
8. Publish results

Ontological approach can facilitate practically all the named steps with the aid of visual mapping schemes which serve as a visionary mind tool and structured framework. Any researcher is in a role of knowledge analyst and he has to describe *concepts* representing entities or `things' within a domain or approaches, tasks, functions, actions, strategies, reasoning processes, etc. Then is the turn of *relations* describing the interactions between concepts or a concept's properties.

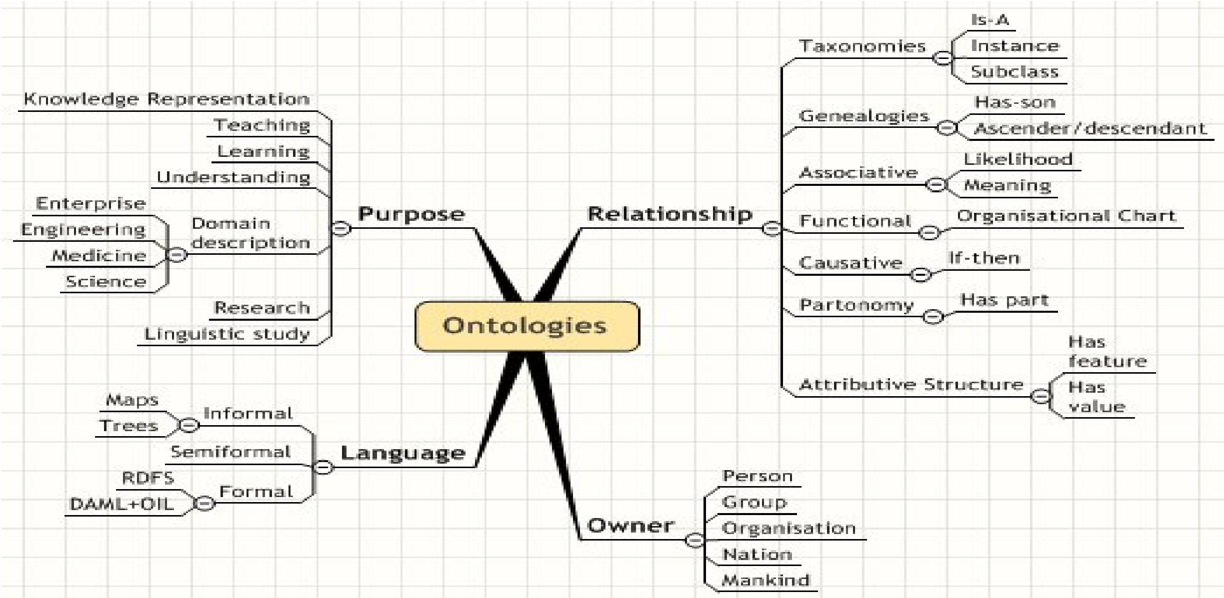

Fig. 1. Ontology Structure

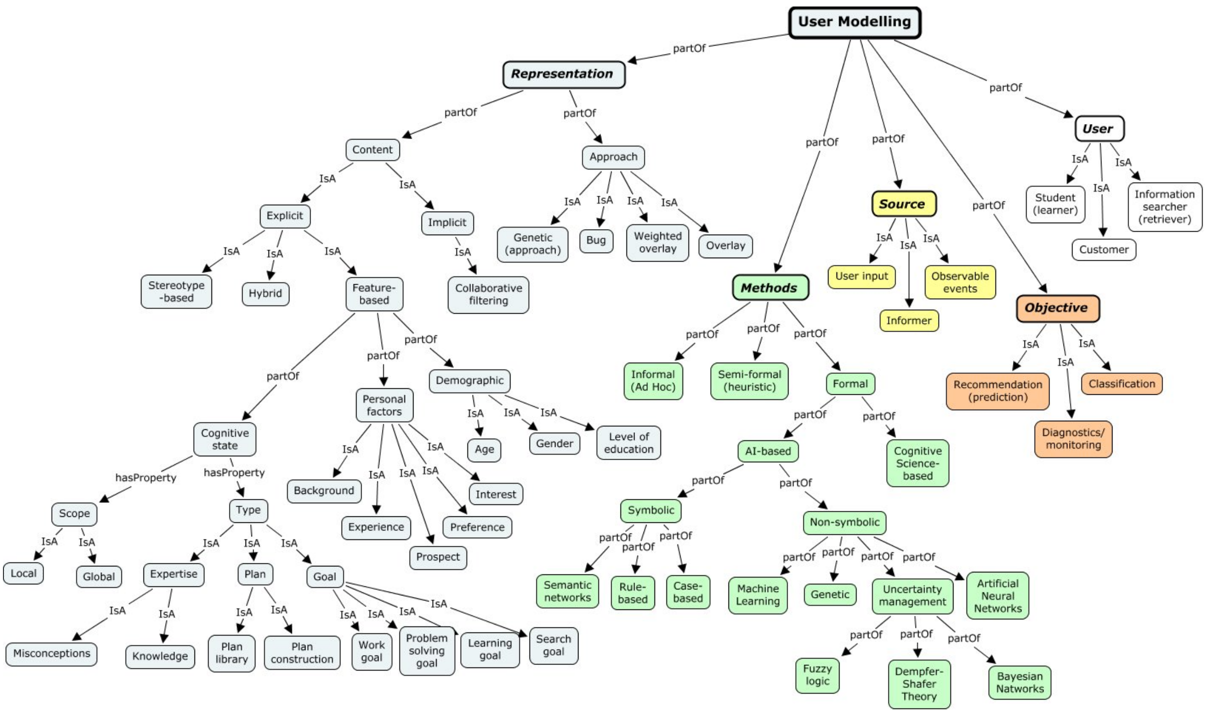

Fig. 2. User Modeling Meta Ontology (UMMO) top levels and one of the branches

Then new approach to conduct systemically more structured research may be proposed. This approach is based on developing of a set of ontologies, e.g.:

- Problem-definition ontology (ontology N1 describing main concepts) and ontology of reviewed approaches (ontology N2 presents genealogy and/or taxonomy describing the history and main branches of the domain problem),
- Experiment framework design (ontology N3 presenting experimental conception) and data structure ontology (ontology N4 presenting input and output data),
- Mathematical modelling and main results ontology design (ontology N5 describing results), etc.

4 MODELING UM META ONTOLOGY

The development of UM Meta Ontology (UMMO) is a part of wider research aimed at development of user model centred learning portal. UMMO is an attempt to externalize the current approaches, techniques, and tools.

Creating ontology is a procedure that cannot be fully automated since ontology development is a rather creative process. But major works in the ontological engineering field deal with syntax problems, not semantics. Until now, only few effective domain-independent methodological approaches have been reported. However, in practice each development team usually follows their own set of principles, design criteria, and steps in the ontology development process.

The current version of UMMO presented in this paper (Fig.2) was developed in part by extracting information about user modelling domain from various sources [1, 3, 8, 9, 12,17, 21,22] and in part by eliciting knowledge of experts (auto-elicitation of co-authors so far). The process of UMMO development was guided by the aforementioned algorithm. At the glossary development stage performed both semi-automatically (keyword extraction) and manually a set of roughly 150 terms was extracted up to this moment. The laddering, generalization, and refinement stages have gone through 10 extensive iterations. A special attention was paid to generalization, since some of the concepts in the UM field have multiple terms associated with them. E.g. behavioural user model, feature-based user model, and individual user model are all synonyms. Monosemic terms were grouped in clusters, the dominated term was chosen at the authors' discretion.

Presented ontology is an upper one that is a hybrid of problem-definition and approach describing ontologies (N1 and N2 from previous paragraph). It is now under re-engineering process and is open to wider discussion with colleagues.

5 CONCLUSION

Ontology development is rather easy for «old» sciences with good structure. Ontological engineering for new, multi-disciplinary and ill-structured disciplines as UM faces a bunch of difficulties in design and development. Ontologies also are rather subjective.

Our paper presents one of the first attempts to show the visionary role of ontologies. They are good for better self-understanding of research and then for knowledge sharing. The challenge of such meta-ontology is not to identify the lower level concepts that correspond to the individual approaches, but to work out and verbalize the meta-level concepts that would help generalize about UM methodology. The role of UMMO is manifold. First, UMMO is an important uniform framework to structure this science field in general. Second, it can be used for teaching UM. Since the field is large and really ill-structured, UMMO is useful as an indexing tool for the learning material.

6 ACKNOWLEDGEMENTS

The work was partially supported by RFBR grant 01-04-00466 and grant from Academy of Finland.

REFERENCES

[1] ACM Computing Classification System http://www.acm.org/class.

[2] Borgo, S., Guarino, N., Masolo, C. Stratified Ontologies: the case of physical objects. In proceedings of theWorkshop on Ontological Engineering. Held in conjunction with ECAI96. – pp. 5-15. Budapest, (1996).

[3] Borst W.N. Construction of Engineering Ontologies. University of Tweenty. Enschede, NL-Centre for Telematica and Information Technology, (1997).

[4] Brusilovsky, P. Methods and techniques of adaptive hypermedia. User Modelling and User-Adapted Interaction, 6 (2-3), 87-129, (1996).

[5] Brusilovsky P., Yudelson M., Gavrilova T. Towards User Modeling Meta Ontology// in Lecture Notes in Artificial Intelligence (LNAI 3538) Ed. Ardissono L., Brna P. & Mitrovic A. Proceedings of 10th International conference User Modeling UM 2005, Springer, 448-452, (2005).

[7] Chandrasekaran, B., Johnson, T. R., Benjamins, V. R. "Ontologies: what are they? why do we need them?". IEEE Intelligent Systems and Their Applications , (1999).

[8] Dolog, P., Nejdl, W. Challenges and Benefits of the Semantic Web for User Modelling. In Proceedings of AH2003 Workshop at 12th World Wide Web Conference, Budapest, Hungary, (2003).

[9] Fischer G. User Modelling in Human-Computer Interaction. User Model. User-Adapt. Interact. 11(1-2): 65-86, (2001).

[10] Gavrilova T., Gorovoy V. Ontological Engineering for Corporate Knowledge Portal Design // In "Processes and Foundations for Virtual Organisations", Eds. L. Camarinha–Matos and H. Afsarmanesh, Kluwer Academic Publishers, 289-296, (2003).

[11]Gruber, T. R. A translation approach to portable ontology specifications. Knowledge Acquisition.Vol. 5, (1993).

[12] Heckmann, D. User model ontology. (Online), retrieved on February 2, http://www.u2m.org/2003/02/ UserModelOntology.daml, (2003).

[13] Jameson, A. Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues. *User Modeling and User-Adapted Interaction, 5*, 193-251, (1996).

[14] Järvinen P. *On Reasearch Methods*. Tampereen Yliopistopaino Oy, Juvenen-Print, Tampere, (2001).

[15] Jenkins A.M. *Research Methodologies and MIS Research*. In Mumford et al, 103-117, (1985)

[16] Jonassen, D.H. Designing constructivist learning environments. In C.M. Reigeluth (Ed.), Instructional design models and strategies, 2nd ed. Mahwah, NJ: Lawrence Eribaum, (1998).

[17] Kobsa, A.: User Modelling: Recent Work, Prospects and Hazards. In: M. Schneider-Hufschmidt, T. Kühme and U. Malinowski, eds., Adaptive User Interfaces: Principles and Practise. Amsterdam: North Holland Elsevier, (1993).

[18] Mizogushi, R. and Bourdeau J. Using Ontological Engineering to Overcome Common AI-ED Problems. *International Journal of Artificial Intelligence in Education, 11*, 1-12, (2000).

[19] Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T., Swartout, W.R. Enabling technology for knowledge sharing. AI Magazine. 12(3), 36-56, (1991)

[20] Noy, N. F. Ontology Development 101: A Guide to creating your First Ontology, Standford University, (2002).

[21] Razmerita, L., Angehrn, A., and Maedche, A. Ontology based user modeling for Knowledge Management Systems. *In Proceedings of the 9th International Conference on User Modeling*, Pittsburgh, USA, 2003, Springer-Verlag, pp. 213-217, (2003).

[22] Rich, E. User Modeling via Stereotypes. In M. Maybury & W. Wahlster (Eds.). *Readings in Intelligent User Interfaces* (pp. 329-342). Morgan Kaufmann Press, CA: San Francisco 117, (1998).

Ambient Audio Notification with Personalized Music

Ralf Jung [1] and **Dominik Heckmann** [2]

Abstract. In this paper we present a user adaptive approach for an ambient audio notification application for multi-user environments. We provide a user centralized notification system working unobtrusively by embedding audio cues in an ambient soundscape by connecting the u2m.org user modeling service. First we introduce the ambient notification system for intelligent environments followed by the identification of extension properties for the user model ontology to get an ambient user adaptive notification service. This short paper should be seen as work in progress that addresses the current research field combining ubiquitous computing and user modeling.

1 Ambient Audio Notification

1.1 Background

In most instrumented environments the visual sense is the primarily used of all human senses. Today‘s user interfaces are focused on visual output for human-computer interaction and ambient media representation [4]. The use of audio signals is limited to simple warning cues and system feedbacks that work in an obtrusive and indiscreet way. Another unwanted side effect in multi-user environments is the distraction of present people from their current tasks. Existing audio notification systems often use sounds of everyday life like birdsongs, wind and rain noises [9]. These audio cues also work in an unobtrusive way but they can not be personalized by the users desire and the claim of privacy is also not taken into account. Other approaches dealing with multi-user environments with respect to an unobtrusive notification transmit the audio signal to the users wireless headphone [10].

1.2 Seamless Notification in Ambient Soundscapes

To prevent the disturbing effect of a traditional notification signal we developed a system that works in a more peripheral and discreet way [2]. We composed and recorded two ambient soundscapes and a jazz standard with respect to some constraints arised from cognitive psychological and musical conditions that have an influence on human perception [3]. Musical factors like rhythm, melody and harmony can have an impact on the listeners behavior. For example, a smooth midtempo ballad song played on a comfortable volume level will have a more calming effect on the listener than a fast and loud rock song. Certainly the instrumention of the soundscapes plays another important role. The arrangement and recording of every single instrument give us the full control of all musical and structural elements. For the composition of the notification melodies we paid attention on several harmonical constraints especially on the well-known *GESTALT*-laws to make sure that the notification signal can easy be recognized by the task user [11].

The system consists of a background soundscape (*core song*) and the optional notification instruments (*audio cues*) that can be selected by the user. For test purposes, we implemented a demo interface in Java that includes a control area in which the composed soundscapes can be selected. The available notification instruments for the selected soundscape can now be matched to the users position in the room overview window. If a registered user shall be notified the system mixes his/her instrument in the current core song and directed the audio cue at the users position. Each instrument can only be selected once to prevent misunderstanding or the users can select group notification where several people are mapped to the same instrument. The efficiency of the system especially the instrument identification rate and the recognition delay were checked in a user test with 28 participants with the result that instrument notification signals were nearly as good as a traditional beep signal [8]. Also the acceptance of the ambient soundscape in the background was high and most of the participants annotated that the soundscapes have a comfortable and calming effect on them. To realize an individual and user designed notification system we identified four components that are interesting for the integration in the user model.

1. Musical Genre.
   The fovorite music style of each user can be represented in the user model to increase the acceptance of the soundscapes (Fig. 1(a)). With that information the system can recommend a soundscape from the sound repository.

2. Musical Instrument.
   Users have the ability to choose their audio cues depending from the previous selected soundscape. Since they have to be able to identify their notification instrument they have to know how it sounds (Fig. 1(b)).

3. Location Information.
   The user model could also provide information about the actual position of each user. That‘s especially important for multi-user environments to guarantee a non-disturbing notification by using a spatial audio system [12] for sending the audio cue near the task user position. The actual indoor position will be calculated on the users PDA by combining user coordinates received by RFID tags and IR beacons [1].

4. Physiological State.
   We also want to consider the emotional state of the user to make sure that the notification signal fits to the actual mental state of the user. We plan to infer this information by monitoring several data, for example heartbeat and blood pressure via biosensors.

[1] Saarland University, Germany, rjung@cs.uni-sb.de
[2] German Research Center for Artificial Intelligence (DFKI), Germany, heckmann@dfki.de

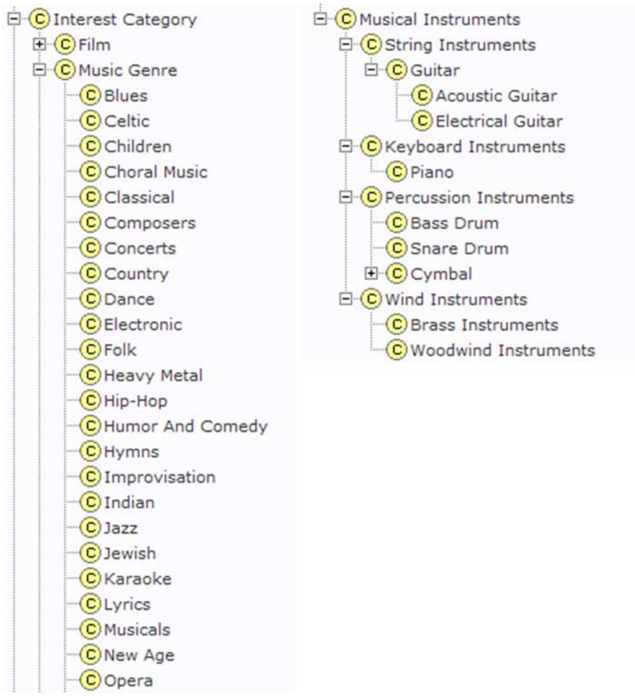

Figure 1. Selected Music Genre (a) and Musical Instruments (b) in the UbisWorld Ontology

2 Making Use of an Application Independent Ubiquitous User Modeling Service

The semantics of the described user model dimensions that are of interest for the ambient audio notification service are defined in the general user model ontology GUMO [7], which is defined in the semantic web language OWL. For our project we apply an application-independent ubiquitous user modeling service, that is based on this GUMO ontology. It manages information about distributed user models on several user model servers, taking privacy issues into consideration. A detailed description of the user modeling system can be found at [6]. The ambient audio notification service can retrieve information from these servers and add new personalized data, like situational statements, to the server. The user models are described in UserML and can be exchanged between user adaptive systems via internet for easy and fast access.

3 User Adaptive Sound Notification

We enable the user to configure his/her personal profile manually with an online user model editor (see *www.ubisworld.org*). However, the adaptive system can realise automatic configuration with the help of the User Model Exchange Services. The following listing shows an example how the adaptive system can retrieve Stefano's interest in the music genre Jazz via a simple HTTP request.

```
http://www.u2m.org/UserModelService.php?
subject=Stefano&auxiliary=hasInterest
&predicate=Jazz
```

In the UserML approach [5] the division of user model dimensions into three main parts is one of the main concepts that has to be achieved. Figure 2 shows the parts auxiliary, predicate and range and three optional meta attributes: situation, privacy and explanation.

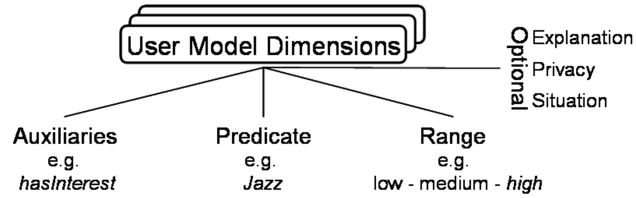

Figure 2. The Structure of User Model Dimensions

The actual user position is calculated on the PDA with the help of Dynamic Bayesian Networks and added via WLAN to the UbisWorld repositories of situational statements. The collected data can be requested by the ambient notification service that mapped the user data to the notification parameters *soundscapes*, *notification instruments*, *spatial audio coordinates* and *audio settings* (Fig. 3). The notification signal can be activated by an occuring event (e.g. incoming email).

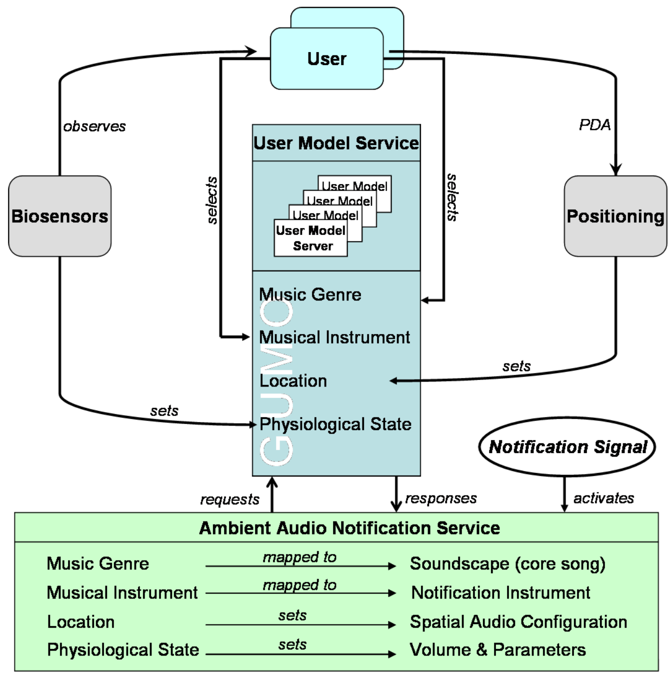

Figure 3. Overview of the *Personalized Ambient Audio Notification (PAAN)* framework

4 Conclusion and Future Work

The main contribution of this short paper is the innovative idea of ambient audio notification with personalized music by enhancing the already existing seamless user notification service with a ubiquitous user model exchange service. The result is a new user adaptive notification service for instrumented rooms that fits the idea of ambient intelligence.

Further work includes the extension of the ambient notification service towards location awareness and context sensitive applications. At the moment we are working on a mobile version to get the users position from an indoor positioning system, then we map these coordinates to the spatial audio framework to get a more discreet and precise notification. One of the task scenarios will be a conference room where the users meet, the system gets their position via PDA and checks periodically the mail account of the user. By receiving an email with an individual preselected keyword in the subject line, the system will configure the spatial audio system to the refreshed coordinates of the task user and mix seamlessly his personal notification instrument in the ambient background sound. The non-disturbing notification with personalized audio cues will be a powerful tool in a lot of daily situations e.g. in waiting rooms on railway stations and airports, examination rooms, museums and presentations.

ACKNOWLEDGEMENTS

This research is supported by the German Research Foundation (DFG) in its Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB 378, Project EM 4, BAIR.

REFERENCES

[1] B. Brandherm and Schwartz T., *Geo Referenced Dynamic Bayesian Networks for User Positioning on Mobile Systems*, Proceedings of International Workshop on Location- and Context-Awareness (LoCA), Munich, Germany, 2005.

[2] A. Butz and R. Jung, *Seamless User Notification in Ambient Soundscapes*, Proceedings of the International Conference on Intelligent User Interfaces, San Diego, California, 2005.

[3] A. Camurri and M. Leman, *Gestalt-Based Composition and Performance in Multimodal Environments*, 495–508, Joint International Conference on Cognitive and Systematic Musicology, Brugge, Belgium, 1997.

[4] A. Dahley, C. Wisneski, and H. Ishii, *Water Lamp and Pinwheels: Ambient Projection of Digital Information into Architectural Space*, Proceedings of the conference on CHI 98, Los Angeles, USA, 1998.

[5] D. Heckmann, *Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource-adaptive computing.*, Proceedings of ABIS2003, Karlsruhe, Germany, 2003.

[6] D. Heckmann, *Situation Modeling and Smart Context Retrieval with Semantic Web Technology and Conflict Resolution*, 34–47, LNAI3946, Springer-Verlag, Berlin Heidelberg, 2006.

[7] D. Heckmann, T. Schwartz, B. Brandherm, and A. Kröner, *Decentralized User Modeling with UserML and GUMO*, 61–66, Decentralized, Agent Based and Social Approaches to User Modeling, Workshop DASUM-05 at 9th International Conference on User Modeling UM2005, Edinburgh, Scotland, 2005.

[8] R. Jung and A. Butz, *Effectiveness of User Notification in Ambient Soundscapes*, Proceedings of the Workshop on Auditory Displays for Mobile Context-Aware Systems (on conference DVD) at Pervasive 2005, Munich, Germany, 2005.

[9] F. Kilander and P. Lönnqvist, *A Whisper in the Woods - An ambient soundscape for peripheral awareness of remote processes*, International Conference on Auditory Display, Kyoto, Japan, 2002.

[10] E. Mynatt, M. Back, and R. Want, *Designing Audio Aura*, 566–573, Conference Proceedings on Human Factors in Computing Systems, Los Angeles, USA, 1998.

[11] M. Reybrouck, *Gestalt Concepts and Music: Limitations and Possibilities*, 57–69, Joint International Conference on Cognitive and Systematic Musicology, Brugge, Belgium, 1997.

[12] M. Schmitz, *SAFIR: A Spatial Audio Framework for Instrumented Rooms*, Proceedings of ITI04: Workshop on Invisible and Transparent Interfaces at Advanced Visual Interfaces, Gallipoli, Italy, 2004.

Ontology-Based User Modeling for Pedestrian Navigation Systems

Panayotis Kikiras[1], Vassileios Tsetsos[2] and Stathes Hadjiefthymiades[2]

Abstract. Human-centered and user-adaptive systems are at the heart of the Design for All and Ambient Intelligence initiatives. Obviously, user models are necessary "ingredients" of such systems. We present a user model for navigation systems (mainly pedestrian), which is based on relevant human wayfinding and navigation theories. We represent this model through a Semantic Web ontology and show how it can be incorporated in an indoor navigation system called OntoNav, which enables personalized path selection.

1 INTRODUCTION

Gluck [1] defines wayfinding as "*the procedure that is used for the orientation and navigating, in order an individual to navigate from one place to another, especially in very huge and complex environments indoors or outdoors*". In general, it is a particularly demanding process, which requires the mobilization of a number of cognitive/mental processes, besides the kinetic ones. Such process is, naturally, executed unconsciously for the majority of people. However, for certain categories of individuals, with certain abilities/disabilities considering their cognitive and/or physical status, wayfinding and navigating may be an extremely cumbersome process. Hence, a "one-size-fits-all" approach does not apply to pedestrian navigation. Personalization of navigation is required and it necessitates the establishment of some appropriate user model that will be taken into consideration when a) computing possible navigation paths, b) selecting the "best" path and c) guiding the user through it by giving her appropriate instructions.

In this paper we present the main theories regarding navigation and their relevance to user models. We exploit such knowledge in order to build a User Navigation Ontology (UNO) that can be used in a navigation system for personalized path selection. Specifically, UNO is an ontology that was developed for modeling users based on their individual characteristics that influence a) navigational decisions (i.e., selection of the optimum path), and b) the form and the means that these navigational decisions are communicated/presented to them. In order to put the presented model in the context of a navigation system we briefly describe OntoNav, an indoor navigation system implemented with Semantic Web technologies.

The organization of the rest of the paper is as follows. In Section 2 we present some theoretical foundations on pedestrian wayfinding and navigation. Additionally, we outline the basic principles and concepts of a navigation-oriented user model. A more formal specification of these concepts is provided in Section 3, where the core of the UNO ontology is presented. In Section 4 we present the basic functionality of OntoNav, while in Section 5 we describe some related work that has partially influenced our work. The paper concludes with directions for future research.

2 MODELING USERS FOR NAVIGATIONAL PURPOSES

2.1 Human Navigation and Wayfinding Theories

Wayfinding is a fundamental human activity and an integral part of everyday life. Individuals are mainly using their knowledge and previous experience with geographic spaces in order to navigate from one location to another. As a result, a huge amount of research literature from the fields of cognitive science, psychology and artificial intelligence examines the mechanisms that enable humans to find their way in unknown and complex environments. In the following paragraphs we discuss the main theoretical approaches to human wayfinding and navigation that have influenced our work.

Wayfinding

Downs and Stea [2] suggested that wayfinding involves the following four steps:

1. Orientation: Finding out where someone is with respect to nearby landmarks and the navigation destination.
2. Route Selection: Selecting a route, under certain criteria, that will eventually lead the individual to the desired destination.
3. Routing Control: Constant control and confirmation that the individual follows the selected route.
4. Recognition of destination: The ability of an individual to realize that she has reached the destination or is located in a nearby area.

In general, the wayfinding ability of individuals is greatly influenced by a number of factors, based on findings from research in human neurophysiology [3]. The most important of these are:

1. Individual Characteristics (e.g., age, sex, cognitive development, perceptual capability, mental and physical condition).
2 Characteristics of the environment (e.g., size, luminosity, signage, utilization, structure, familiarization with it).
3. Learning Processes (e.g., learning strategies, learning conditions, learning abilities).

Furthermore, the wayfinding ability of individuals is mainly affected by the following four factors: spatial ability, fundamental information processing capabilities, prior knowledge of the environment and motor capabilities. S*patial ability* can be defined as the ability of every individual to perceive the surrounding environment with its sensing and cognitive mechanisms. This ability includes all cognitive procedures that are used whenever we are learning our environment and comprehend correlations among its elements. This leads to *spatial consciousness*, which describes the degree to which an individual understands/reacts with the environment using her spatial ability. Thus, wayfinding is a dynamic and demanding cognitive procedure, which involves many spatial and navigational abilities. Moreover, similarly to every other human activities, not every individual has the same navigational skills

[1] Dept of Computer & Communications Engineering, University of Thessaly, Greece. Email: kikirasp@inf.uth.gr
[2] Pervasive Computing Research Group, Dept of Informatics & Telecommunications, University of Athens, Greece, Email: {b.tsetsos, shadj}@di.uoa.gr

[4]. This fact calls for a classification of potential users of a navigation system so that it could provide its services in a way tailored to their specific cognitive and physical abilities/disabilities.

Navigational Awareness

Navigational awareness is defined as the wayfinding task which takes place when the individual who navigates in an area has complete knowledge of the navigation environment. There are two distinct types of navigating through an environment, with significant differences between them. The first navigation type is based on what is called *procedural* or *route knowledge*. The procedural knowledge is human centered (ego-referenced) and is mainly acquired through personal exploration of an unknown environment. The main characteristic of the procedural knowledge is that, while an individual can navigate from one landmark to another in a known route, she has no other knowledge about alternatives routes (fastest, quickest, etc.). The second type of navigation is based on the *survey knowledge*. Such knowledge is acquired through iterative multiple explorations of an area following different path each time. This type of survey knowledge is characterized by its ability to support distinctive places of the environment (landmarks) as reference points and, thus, is called world-referenced.

Research in this area has shown that acquiring complete knowledge of unknown, big and complex areas is a dynamic process, which involves four distinct steps [5]:

1. Recognition of landmarks: Objects may constitute landmarks for two reasons a) for their distinguishing characteristics, and b) due to their individual significance [6]. Objects can be distinguishable because of their architectural style, size, or color [7]. Moreover, objects can become significant landmarks whenever they provide navigational information (e.g., when they are positioned at a crossroad or junction, at big interior halls that connect different corridors, etc.).

2. Correlation of routes or connections with landmarks: Routes and connections are formed while navigating between two landmarks. Acquiring route knowledge is highly correlated with the process of recognizing landmarks, which can be recalled with the same cognitive mechanism that is used to recall a route at a future time. This step is the cognitive procedure of matching routes with landmarks.

3. Primary Survey Knowledge: This type of knowledge is acquired after a thorough survey and exploration of the navigation environment. When acquired, it provides the means to calculate different routes and to estimate the distance between landmarks.

4. Area–Route Segmentation: This step provides the mechanisms to decompose a huge area to smaller segments/regions. Such smaller regions are parts of bigger regions, which in turn form other bigger ones and so on. This "segmentation procedure" enables the individual to mentally focus on regions relevant to its navigation task, to discover relations between different spaces, and, thus, by minimizing the amount of information to be processed optimizes the navigating performance of an individual.

2.2 Navigation-oriented User Modeling

According to the previously presented theoretical findings, a navigation-oriented User Profile (UP) is based on attributes from the following categories/components (see Figure 1):

1. General User Demographics: This category captures all the basic user information such as name (required only for user identification and profile indexing, thus it can simply be a nickname), age, gender, as well as a series of optional information, e.g., communication details, etc. (if required by the application for billing, statistical or other reasons).

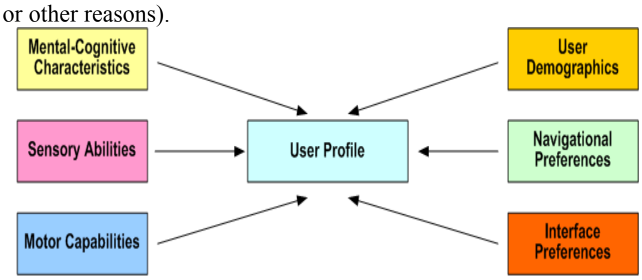

Figure 1. Components of a navigation-oriented User Profile

2. Mental/Cognitive Characteristics: this category captures all information considering user's mental/cognitive abilities as follows:

i. *Consciousness functions*: in this Boolean attribute the system captures the existence of possible malfunctions in the user consciousness abilities. Such abilities correspond to general mental functions which control user's state of awareness and alertness.

ii. *Orientation disability*: This Boolean attribute captures user's orientation ability, which corresponds to knowing and ascertaining her relation to oneself, to others, to time and to the surrounding environment. This ability describes the cognitive abilities that an individual must possess in order to be able to navigate in a geographical space. Hence, potential malfunctions in this ability significantly hinder the navigation procedure.

iii. *Mental disabilities*: This Boolean attribute holds true if the user has disabilities considering her mental functions (mental impairment, Alzheimer disease, etc.).

iv. *Mental functions considering user's behavior and personality*: In this subcategory the system captures behavioral and personality characteristics such as introversion–extroversion, social abilities, psychic and emotional stability. These characteristics differentiate one person from another and this knowledge is used for the personalization of the routing instructions. As discussed in [9], such information affects the way that an individual comprehends and follows routing instructions.

v. *Concentration to an objective*: The World Health Organization defines this mental function as "*the mental ability of an individual to remain focused on an external stimuli or an internal experience for a certain period of time*". Difficulty on this function is more often met in elderly people, teenagers and children.

vi. *High level cognitive functions*: this category considers difficulties in high level cognitive functions, such as decision making, planning and execution of actions and plans, degradation of memory functions, etc. Potential malfunction of any of these cognitive functions may lead to difficulties for the users to understand and execute complex instructions in a timely manner. Therefore, a navigation system should be able to correspond to such information by selecting proper paths and customizing the routing instructions in a way suitable for a user suffering from such impairments.

3. User's Sensory Abilities: Sensory impairments affect the way a user exploits her sensing abilities (especially viewing and hearing) during wayfinding. This category is further divided to two subcategories: visual and audile abilities. The visual abilities of users can be categorized using the following main criteria:

i. Visual Sharpness: A: perfect, -B: good, -C: medium, -D: bad.

ii. Visual Quality: Impairment in this ability affects the way an individual perceives light, color, texture, contrast and, in general, the quality of user's vision. Possible quality values are – A: perfect, -B: good, -C: medium, -D: bad.

The audile abilities of users are divided in four categories – A: perfect, -B: good, -C: medium, -D: bad, (where A means that the user has full hearing ability and D that she cannot hear at all).

4. User's Motor Abilities: This category captures a user's ability to move from one place to another with respect to the way she controls and coordinates her movement. Motor abilities refer to all kinetic abilities of users and not only to those associated to their mobility, although the latter are more important from the perspective of navigation. Users are categorized as having:

i. Autonomous mobility without assistive devices

ii. Mobility supported by an escort (with or without assistive devices).

iii. Autonomous mobility with wheelchair.

iv. Autonomous mobility with assistive devices (other than wheelchair)

Note that the user profile of a user supported by an escort should be the profile of the escort, since the latter is responsible for the navigation of the disabled user.

5. Navigational Preferences: This category captures user's navigational preferences. Typical preferences are:

i. No specific preferences.

ii. Selection of the shortest route first.

iii. Selection of the fastest route first.

iv. Preference in most "popular" path elements (e.g., main corridors and stairs).

v. Avoidance of stairs.

vi. Avoidance of crowded areas (e.g., for blind users).

vii. Selection of the most/less popular path among all users.

viii. Existence of landmarks in computed paths.

ix. Dynamic tracking during navigation and provision of routing corrections.

6. Interface Preferences: This category captures user's preferences considering the means and the media in which user will receive routing instructions:

i. Type of user's device (e.g., PDA, mobile/smart phone, mobile computer, information kiosk).

ii. Modality of instructions' presentation:

a. Only textual information

b. Both textual and visual information

c. Only visual information

d. Both textual and audio information

e. Both visual and audio information

f. Only audio information.

2.3 Discussion

As it is obvious from the above categorization, a UP is defined as the set of the characteristics chosen by the user. Every UP attribute takes either a value from a category of values or a Boolean value (Yes/No or True/False). Additionally, some attributes may assume values from a closed set (e.g., good, bad, etc.).

Apart from the aforementioned components that affect navigation-oriented user modeling, special emphasis should be given to the factors *age* and *gender*, since many of the abovementioned human navigational and wayfinding capabilities are dependent on them [8][9]. Moreover, gender and age affect the way that routing guidelines should be presented to users [10][11]. For example, for male users in the age range 16-65 the most suitable way of providing routing instructions is by using descriptions in metric and geographic notations (e.g., "follow this route to the north for one kilometer, then turn towards north-east and drive for about two more kilometers"). On the other hand, for female users, irrespective of their age, the most suitable way for providing navigational instructions is by using landmarks (e.g., "follow this road until you arrive to the next church, then turn right until you arrive at a square, then you may find your destination at the upper part of the square"). For children and elderly people the most suitable way of providing routing instructions is by segmenting the path in many easy-to-remember segments, i.e., having at least one clearly distinguishing landmark.

The aforementioned age and gender categorization is also applicable to the user interface modality used for presentation of the routing instructions. Therefore, for males the best choice is audio instructions. On the other hand, for females the most efficient modality is visual representations of landmarks with textual or auditory instructions. For elderly and young people the visual representation of landmarks is the most effective approach, in combination with maps with arrows pointing at the desired destination.

3 A USER NAVIGATION ONTOLOGY

The model described in the previous section has to be specified in a suitable form (possibly Web-based) in order to be used in modern applications. Hence, we have decided to represent it through a Semantic Web ontology. For that purpose we have used the Web Ontology Language (OWL) [12] for describing the user classes and their properties. Ontology-based systems are becoming more and more popular due to the inference and reasoning capabilities that ontological knowledge representation provides. Moreover, Semantic Web standards, and technologies in general, provide a solid basis for open and interoperable information systems.

For the development of the UNO ontology we followed the directives of ontology engineering that promote ontology reuse and alignment between existing ontologies. Specifically, during ontology development we have tried to extend some of the concepts specified in the GUMO ontology (see section "Related Work"). An extract of the UNO concept hierarchy is shown in Figure 2, while Figure 3 illustrates the basic UNO properties. Informal definitions of the top-level UNO concepts follow (the definitions of properties are regarded straightforward):

Ability: the super-class of the various abilities of a user with regard to the navigation procedure. A user may have many abilities. Disabilities may be defined through the use of the Quality class values (see below).

Demographics: value classes for user demographics (age, gender). Its subclasses are implemented as value partitions as dictated by the W3C Semantic Web Best Practices Group [18].

Quality: another class representing a value set for describing the degree/quality of the various abilities. Its values are {bad, medium, good, perfect}. A bad quality value for an ability denotes a disability.

User: an abstract class that subsumes the more specific defined user classes.

The main difference between UNO and GUMO, apart from their scope, is that UNO is used actively in inference procedures, while GUMO provides a core knowledge base (i.e., taxonomy and assertions of individuals) for basic classification of users and their characteristics. Hence, a key feature of UNO lies in the formal definition (through restrictions, and necessary and sufficient conditions) of user classes. In the current version of UNO we have included a minimal set with some possible classes. Each specific navigation application should extend this set appropriately. The use of the OWL-DL language enables very expressive user definitions. Indicative definitions (in mixed OWL and first-order-logic-like notation, for readers unfamiliar with Description Logics notation) of such *defined concepts* are:

YoungWheelchairedUser ↔
∃ hasAbility AutonomousWheelchairedMobility ∧
∃ hasAge LessThan18

VisuallyImpairedMaleAdultUser ↔
∃ hasAbility (AbilityToSee ∧ hasValue(hasQuality, bad)) ∧
∃ hasAge Between18and60 ∧ hasValue(hasGender, male)

(Note: *hasValue* is a reserved OWL term)

After performing reasoning on an ontology with such defined user classes, these will be classified under the generic User class and the various user instances will be classified accordingly.

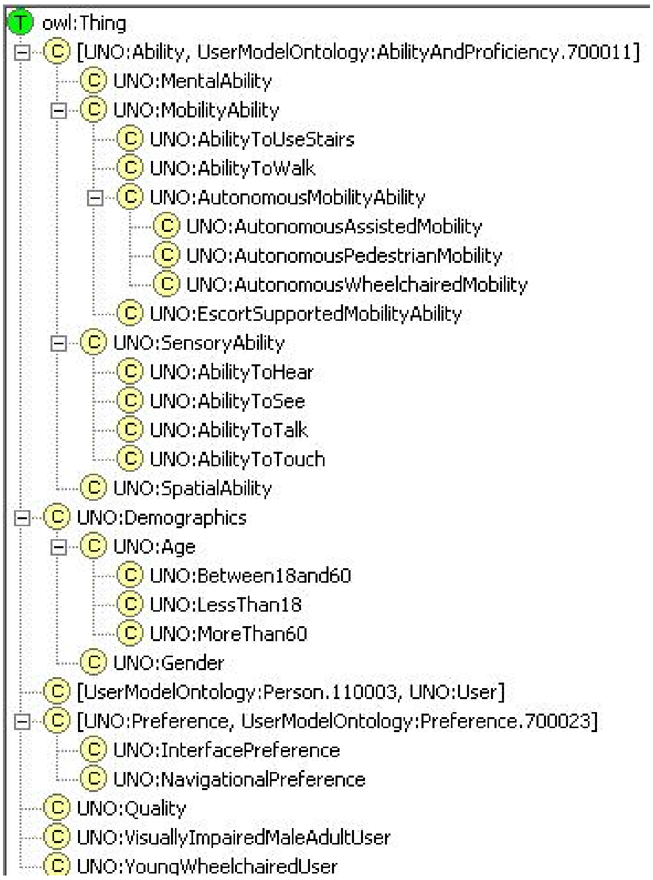

Figure 2. The basic UNO taxonomy (the prefix *UNO* denotes a UNO class, while *UserModelOntology* denotes a GUMO class)

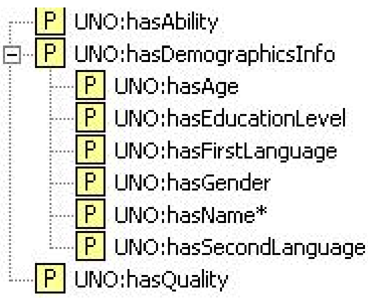

Figure 3. The basic UNO hierarchy of properties

Regarding alignment with GUMO, some UNO classes are declared as equivalent to GUMO classes (e.g., Preference). Moreover, some individuals of GUMO have been transformed to primitive classes in UNO (e.g., individual AbilityToTalk of GUMO class AbilityAndProficiency has been asserted as class AbilityToTalk in UNO). Regarding demographics information, we have modeled some relevant GUMO instances as binary properties, since otherwise we would have to create a different instance of such information for each separate user. The aforementioned transformations (instances to classes and instances to binary relations) have been performed in order to enable more complex concept expressions for describing user class. Finally, we should note that there are GUMO classes that have not incorporated/aligned to the current version of UNO, although they are relevant to the domain of navigation. For example, the class Motion could be used for supporting dynamic tracking and route corrections and the class PhysicalEnvironment could support the context-aware adaptation of navigation instructions (e.g., high noise level could trigger increase in the volume level of audio instructions).

4 OntoNav: A HUMAN-CENTERED INDOOR NAVIGATION SYSTEM

In this section we describe how the aforementioned UNO ontology is incorporated in an indoor navigation system called OntoNav. Note that in the description of OntoNav we focus on issues that assist the reader in understanding how UNO affects the navigation procedure. More details on OntoNav design and implementation can be found in [13][14].

OntoNav, is an integrated indoor navigation system, which is based on a hybrid modeling (i.e., both geometric and semantic) of such environments. OntoNav is purely user-centric in the sense that both the navigation paths and the guidelines that describe them are provided to the users according to their physical and perceptual capabilities as well as their particular routing preferences. For the description of path elements (e.g., corridors, junctions, stairways) an *Indoor Navigation Ontology* (INO) has been developed. The instances of such ontology are created by annotated GIS building blueprints. In order to compute the candidate paths for a specific user request, a dual graph representation of the ontology is also created (*topology graph*). The main OntoNav components along with the main workflow are shown in Figure 4.

The basic functionality of OntoNav can be summarized in the following steps:

1. Creation of a User Profile (if the user is unknown to the system, retrieval of a cached one else). In terms of ontological knowledge management, UP creation is the process of asserting UNO (concept and property) instances about the user, her abilities and demographics.
2. Invocation of the Navigation service where the desired destination is given as input to the system.
3. Creation of a user-compatible topology graph (i.e., that can be traversed by the user). This task is performed by applying production rules to the UP information (UNO instances) and the path elements semantics (INO instances).
4. Computation of the k-Shortest Paths between origin and destination locations in this graph.
5. Ranking of these paths according to additional UP information and selection of the "best" path for the specific user along with the most appropriate instructions for this path.

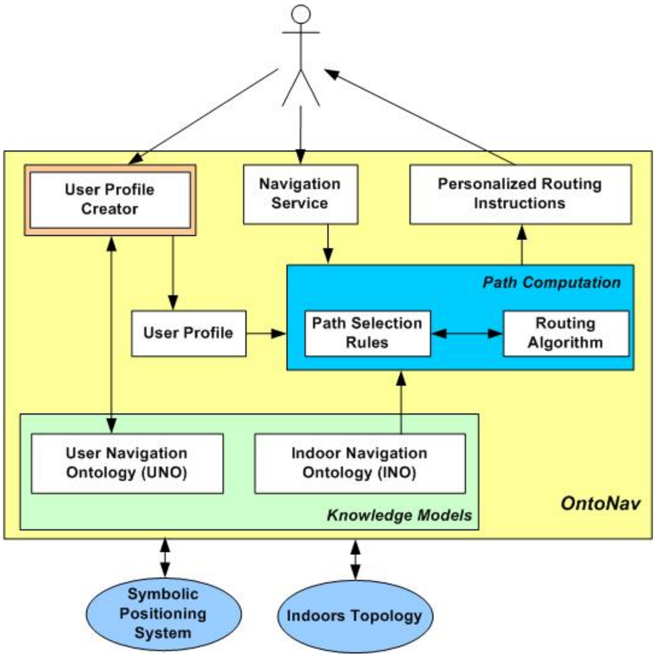

Figure 4. OntoNav architecture and basic workflow

The path-selection process (steps 3 and 5) is performed through sets of production rules. The definition of such rules involves both the spatial semantics (expressed through INO) and the user semantics (expressed through UNO). The rules are applied to the INO instances in order to infer and assert which paths are considered accessible and appropriate for each user request. Such path-selection rules are further analyzed to *physical navigation rules*, *perceptual navigation rules* and *navigation preferences*. The physical navigation rules are applied first (step 3), in order to discard any paths that are not physically accessible by the user. The perceptual navigation rules depend on the user's cognitive/mental status, demographics (e.g., age, education) as well as sensory abilities. Finally, paths that match the user preferences (e.g., paths containing elevators) are identified with the application of the navigation preference rules. The rules are described through the Semantic Web Rule Language-SWRL [15]. Some indicative rules are the following (the UNO user classes used in these rules are hypothetical and their definitions are analogous to those presented in Section 3):

Rule 1 (Physical Navigation Rule)

UNO:HandicappedUser(u) $\wedge$ INO:Stairway(s) $\rightarrow$
INO:isExcludedFor(s,u)

Rule 2 (Perceptual Navigation Rule).

UNO:BlindUser(u) $\wedge$ INO:hasDescription(pass,descr) $\wedge$
INO:Textual_Description(descr) $\rightarrow$
INO:hasPerceptualPenaltyFor(pass,u)

Rule 3 (Navigation Preference).

UNO:LazyUser $\wedge$ INO:Motor_Passage(p) $\rightarrow$
INO:hasPreferentialBonusFor(p, u)

As one can observe, some of these rules "mark" the path elements that should be excluded from the user-compatible topology graph (through the isExcludedFor property), while others reward/penalize some path elements (through the properties hasPreferentialBonusFor, hasPerceptualPenaltyFor, etc.). The final ranking of the traversable paths (step 5) is based on such bonus/penalty assertions and on the path length, which always remains a key selection criterion.

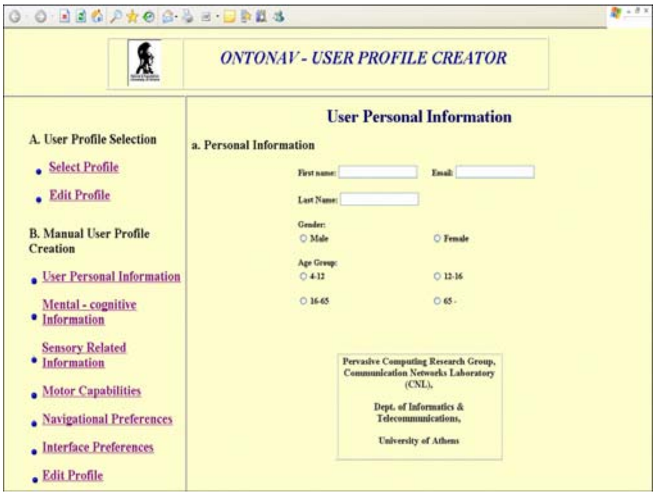

Figure 5. Screenshot of the User Profile Creator

4.1 User Profile Management in OntoNav

A key component of the OntoNav architecture, with respect to user modeling, is the User Profile Creator (UPC). This component provides users with an interface that enables them to create their UP according to the UNO terminology (see Figure 5). The first time a user invokes the system's interface, she has the option to choose a profile from a set of predefined UPs. Currently there are four general types of UPs representing:

a. users without disabilities,
b. users with motor disabilities,
c. users with hearing disabilities, and
d. users with visual disabilities

The user can even choose a combination of the latter three UP types and subsequently customize such predefined profile accordingly. Alternatively, she may create a custom profile by providing all the indispensable information that can describe her physical and cognitive status, as well as her navigational and interface preferences. Moreover, the UP is completely dynamic; the user may view, alter, update, or delete part or all of her profile if necessary.

The OntoNav architecture also specifies a navigation supporting component called Navigation-Aiding Module (NAM) (see Figure 6 and refer to [13] for more details). Its primary task is to detect deviations from the initially planned path and help users return to it or find a new more suitable one. Since NAM continuously tracks the users' navigational behavior (in terms of spatiotemporal changes) it could be exploited as a means of inferring or "calibrating" (i.e., correcting) some UP elements. Inference of UP elements is a hot topic in UP creation since users are not always willing to explicitly describe their profile. Moreover, they are often reluctant to reveal any disabilities they may have. Hence, an unobtrusive monitoring systems, such as NAM, could facilitate seamless UP creation. Such functionality would involve sophisticated pattern matching algorithms (UP Inference and Calibration component in Figure 6) This component, currently under development, tries to infer user characteristics from the trajectories she follows during navigation and her navigation-relevant history (user movement statistics). However, such inference demands accurate indoor positioning systems which are not widely available and deployed yet.

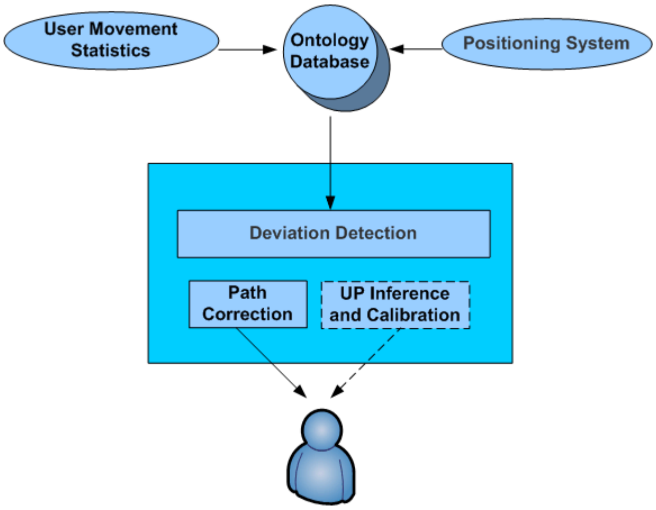

Figure 6. The Navigation Aiding Module

5 RELATED WORK

To our knowledge there is no other user model for describing user characteristics from the perspective of navigation. On the other hand, there are some generic, user modeling efforts that try to cover a wide range of application domains and to adopt open technologies for enabling interoperability between systems. The most relevant work of this category is the General User Model Ontology (GUMO) [16]. GUMO has means of representing several "user dimensions" such as user demographics, user abilities, user emotional and psychological status, etc. In addition, it supports the specification of some auxiliary information such as the preferences, interests, and knowledge of the users. The main advantage of GUMO is that it is implemented in OWL, which has become very popular in the Semantic Web [12] community. This language not only provides a well-defined syntax for user models but is also capable of describing the semantics that are implied by a model. As already mentioned, we have tried to align UNO with GUMO by reusing and extending all suitable concepts and attributes.

GUMO has been partially influenced by the UserML language [17]. UserML's objective was to provide a commonly accepted syntax, based on the XML standard, for representing user models in Web applications. UserML is quite generic and, thus, can be used as a syntax layer for any semantic user model.

6 CONCLUSION AND FUTURE WORK

In this paper we have presented some background knowledge on navigation theory from various disciplines (e.g., psychology, physiology), which directly affects any navigation-oriented user model. Furthermore, we have taken into consideration these theoretical implications in order to construct a user ontology. Finally, we have shown how such ontology is instantiated and actively involved in the navigation procedure of the OntoNav system through inference rules.

However, several issues remain open for further research in this area. One of the most interesting and important issues is the (semi-) automatic user model creation. Specification of rules that represent dependencies between model entities (derived from relevant theories) seem to be a promising solution, although hard to implement. For example, the ability of a user to concentrate on an objective may be automatically inferred by her age. Another challenging issue, and "common" with respect to user profiles, is privacy protection (since UNO describes also personal information such as health/physical/mental status). Finally, as UNO is still under development, we have not taken into consideration all the UP components identified in Section 2.2, since some of them are difficult to capture (e.g., mental/cognitive characteristics).

ACKNOWLEDGEMENT

This work was performed in the context of the "PENED" Programme, co-funded by the European Union and the Hellenic Ministry of Development, General Secretariat for Research and Technology (research grant 03ED173).

REFERENCES

[1] Gluck, M., "Making Sense of Human Wayfinding: A Review of Cognitive and Linguistic Knowledge for Personal Navigation with a New Research Direction". Syracuse University Technical Report 3, (1990).
[2] Downs,R.J., and Stea, D., "Cognitive Maps and Spatial Behavior". Image and Environment, 8-26. Chicago: Aldine Publishing Company, (1973).
[3] Allen G. "Spatial Abilities, Cognitive Maps, and Wayfinding - Bases for individual Differences in Spatial Cognition and Behavior". in: R. Golledge (Ed.), "Wayfinding Behavior - Cognitive Mapping and Other Spatial Processes". pp. 46-80, Johns Hopkins University Press, Baltimore, (1999).
[4] S. Timpf, G. Volta, D. Pollock, and M. Egenhofer "A Conceptual Model of Wayfinding Using Multiple Levels of Abstraction". in: A. Frank, I. Campari, and U. Formentini (Eds.), "GIS—from Space to Theory: Theories and Methods of Spatio-Temporal Reasoning", Pisa, Italy, Lecture Notes in Computer Science 639, pp. 348-367,
[5] Siegel, A.W., White, S.H., "The Development of Spatial Representation of Large-Scale Environments". In H.W. Reese (Ed.), Advances in Child Development and Behavior. New York: Academic Press (1975).
[6] Lynch, K., "The Image of the City". Cambridge, MA: MIT Press (1960).
[7] Weisman, J. "Evaluating architectural legibility: Wayfinding in the build environment". Environment and Behavior, 13, 189-204. (1981
[8] Sigrid Schmitz, "Gender Differences in Acquisition of Environmental Knowledge Related to Wayfinding Behavior, Spatial Anxiety and Self-Estimated Environmental Competencies", Sex Roles, Vol. 41, No. 1/2, (1999).
[9] kathleen Kirasic, "Age differences in adults' spatial abilities, learning environmental layout, and wayfinding behavior", Spatial Cognition and Computation 2: 117–134, 2000.
[10] Carol A. Lawton and Janos Kallai, "Gender Differences in Wayfinding Strategies and Anxiety About Wayfinding: A Cross-Cultural Comparison", Sex Roles, Vol. 47, Nos. 9/10, November 2002
[11] Martin Raubal and Stephan Winter "Enriching Wayfinding Instructions with Local Landmarks", in: M.J. Egenhofer and D.M. Mark (Eds.): GIScience 2002, LNCS 2478, pp. 243–259, 2002.
[12] Antoniou, G., & Harmelen, F.v. "A Semantic Web Primer". Massachusetts: The MIT Press, (2004).
[13] Tsetsos V., Anagnostopoulos C., Kikiras P., Hasiotis T., and Hadjiefthymiades S., "A Human-centered Semantic Navigation System for Indoor Environments", IEEE International Conference on Pervasive Services (ICPS'05), Santorini, Greece, (2005).
[14] OntoNav Web site, URL: http://p-comp.di.uoa.gr/projects/ontonav/index.html
[15] Horrocks, I., Patel-Schneider, P., Harold, B., Tabet, S., Grosof, B., & Dean, M., "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", World Wide Web Consortium Member Submission, http://www.w3.org/Submission/SWRL/, (2004)
[16] Heckman, D., Brandherm, B., Schmitz, M., Schwarz, T., von Wilamowitz-Moellendorf, B.m., "Gumo – the general user model ontology", User Modeling (UM), Edinburgh, Scotland, (2005).
[17] Heckmann, D., Krüger, A. "A User Modeling Markup Language (UserML) for Ubiquitous Computing", User Modeling (UM), (2003).
[18] W3C Semantic Web Best Practices and Deployment Working Group, www.w3.org/2001/sw/BestPractices/, (2001)

Exploiting the Link Between Personal, Augmented Memories and Ubiquitous User Modeling

Alexander Kröner and **Dominik Heckmann** and **Michael Schneider**[1]

Abstract. Dense logging of a user's interactions with an intelligent environment enables the creation of artificial memory structures, which augment the user's natural memory. In this article we discuss the interrelationship between such augmented memories and ubiquitous user modeling, and the particular benefits of linking both concepts. On the basis of a prototype implementation, we illustrate how ubiquitous user modeling contributes to the building of personal, augmented memories, and discuss how the latter one may enrich ubiquitous user models and thus grant other applications access to memory content. We conclude with a discussion of how users may exploit such hybrid systems to share their personal memories with other users.

1 INTRODUCTION

Driven by promising applications in the field of user modeling and decision support, research on mechanisms for augmenting a user's natural memory with information automatically captured in the user's context has been conducted since many years (see for instance [2], [5], [6]). Following that tradition, the project SPECTER researches how a personal, augmented memory can be built and exploited in an everyday scenario, in particular, shopping (cf. [4]). SPECTER's event-based memory is created from dense logs of data automatically captured in its user's context. In this article we describe how ubiquitous user modeling may contribute to building and exploiting this specific kind of data structure.

2 BUILDING AUGMENTED MEMORIES

The basis of augmented memories in SPECTER are *perceptions*, which are built from sensor data recorded in so-called *RDF:Stores*. In brief, these can be described as lightweight sensor memories, which implement based on a unique interface for each sensor specific storage and access mechanisms. RDF:Stores allow applications to access sensor information via a push as well as a pull service. This allows an efficient communication between SPECTER and the environment: Whenever SPECTER enters a new environment, it can poll all information available from the sensors' RDF:Stores at once via the pull mechanism. Later, SPECTER will be automatically notified about changes in the environment via the push mechanism.

These "sensory" memories feed a short-term memory with perceptions. There, incoming perceptions undergo a plan-based situation recognition mechanism, which initiates the construction of entries for the long-term memory and triggers situated user support.

[1] German Research Center for Artificial Intelligence (DFKI), 66123 Saarbrücken, Germany, email: firstname.lastname@dfki.de

This way of building augmented memories can be complemented with ubiquitous user modeling: its centralized, uniform structure allows SPECTER to exploit other application's sensing and processing capabilities in order to build entries in the augmented memories and to determine the user's current context. The user benefits from such linking as well, since feedback provided to the ubiquitous user model becomes accessible to SPECTER.

For example, the personalized ambient audio notification (PAAN) service for intelligent environments models the user's preferred musical genres in order to generate personalized music (cf. [1]). This allows to discretely inform the user about private events in public places. These music genre preferences are shared with U2M, and can be edited by the user in the U2M user model editor. SPECTER may apply these preferences in order to adapt retrieval processes performed on the augmented memory.

3 EXPLOITING AUGMENTED MEMORIES

Augmented memories may serve diverse purposes, for instance, context-based reminding and recommending for decision support (cf. the *recomindation* paradigm described in [7]). Despite a clear focus on personal use, it might be reasonable to share parts of the personal memory with other users or application—for instance, the user may select favorite items from the augmented memories, and provide these as examples to the environment in order to receive links to similar items offered there. Such communication may range from submission of somewhat neutral items (e.g., some product seen without personal rating) to complex descriptions of situations representing confidential user actions and/or personal options.

This information has to be delivered to interested applications, which may not only process them for serving the user/system's request, but also for building their own user model. Now the augmented memory described in this article resembles a specific kind of user model on its own, which means that the whole communication is a typical ubiquitous user modeling process. Consequently a ubiquitous user modeling platform qualifies for realizing the exchange; this platform has on the hand to protect privacy and application constraints, and on the other hand to provide any external application with access to the augmented memory.

This procedure provides a straightforward approach for sharing memories between a user and diverse ubiquitous computing applications; in addition, it opens the way for sharing memories between users. The great success of blogging (cf. [8]), shared knowledge sources (e.g., Wiki), collaborative tagging (e.g., *del.icio.us*) and similar applications sometimes associated with the notion Web 2.0 is an impressive demonstration of the high demand for such exchange between users. Sharing personal, augmented memories by means of

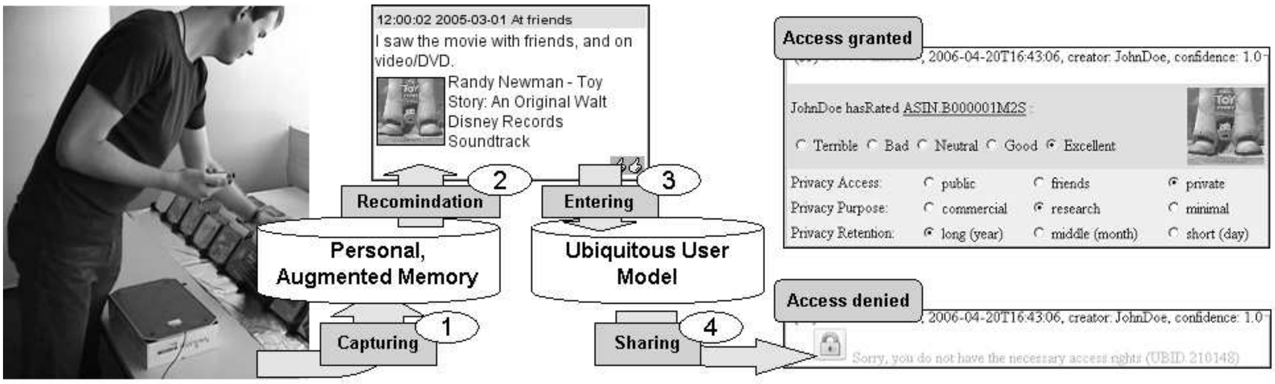

Figure 1. The artificial memory records CDs the user looked at. The user may later reflect on these experiences, and publish selected parts in an ubiquitous user model where it becomes accessible to external applications and other users.

ubiquitous user modeling is one way to transfer such ideas to ubiquitous computing.

Figure 1 depicts this idea by means of the applications SPECTER and U2M. Events observed in the user's environment are automatically captured in SPECTER's personal memory (1). SPECTER exploits such information for recomindation—in our example, the event is shown where the user encountered a certain audio CD the first time (2). At any time the user may decide to submit such information to the ubiquitous user model realized by U2M (3). In our example, the user shares his ratings of audio CDs he discovered in the store. U2M makes the data accessible to other users with respect to the user's privacy preferences (4). The user may specify these for the current sharing process (and thus attach situational constraints), or rely on U2M's default reasoning which derives sharing preferences for the involved objects, actions, etc. from personal defaults set in the ubiquitous user model (cf. [3]).

Future work. Our example illustrates a sharing process that includes activities performed manually by the user. While this approach provides the user with precise control over the shared memories, it requires an amount of attention the user will not be able (or willing) to devote to sharing during other activities. Now the user could specify default preferences on privacy, trust and other aspects of sharing in preparation of such activities, but this requires precise planning of future actions which is not only cumbersome, but also cannot take into account unexpected sharing opportunities.

Automated and/or assisted sharing of personal, augmented memories in ubiquitous computing environments is a way to handle such issues. It forms the background of research in the recently started project SHAREDLIFE; in a multi-user scenario we plan to explore various questions related to sharing memories, including:

- How can community structures serve the situated lookup of information?
- How can the users' sharing behavior be exploited to strengthen communities?
- How can personal sharing preferences be specified using reflection on past events?

4 CONCLUSION

In this article we discussed how personal, augmented memories created from perceptions in an intelligent environments relate to ubiquitous user modeling. On the basis of a prototype implementation, we showed how ubiquitous user modeling may contribute to the building of such personal memories, and how these may exploit a platform for ubiquitous user modeling in order to share memory content with other applications and users. The latter process will be in the focus of our future activities, which will address issues related to automated and semi-automated sharing of personal memories.

ACKNOWLEDGEMENTS

This research was supported by the German Ministry of Education and Research (BMBF) under grant 01 IW C03 (project SPECTER) and 01 IW F03 (project SHAREDLIFE) as well as by the German Research Foundation (DFG) in its Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB 378, Project EM 4, BAIR. We would like to thank all project members for their valuable contributions.

REFERENCES

[1] Andreas Butz and Ralf Jung, 'Seamless user notification in ambient soundscapes', in *Proceedings of IUI 2005*, San Diego, California, USA, (January 2005).

[2] Jim Gemmell, Lyndsay Williams, Ken Wood, Roger Lueder, and Gordon Bell, 'Passive Capture and Ensuing Issues for a Personal Lifetime Store', in *Proceedings of The First ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE '04)*, pp. 48–55, New York, USA, (2004).

[3] Dominik Heckmann, *Ubiquitous User Modeling*, Ph.D. dissertation, Department of Computer Science, Saarland University, Germany, 2005.

[4] Alexander Kröner, Dominik Heckmann, and Wolfgang Wahlster, 'Specter: Building, exploiting, and sharing augmented memories', in *Workshop on Knowledge Sharing for Everyday Life 2006 (KSEL06)*, ed., Kiyoshi Kogure, pp. 9–16, Kyoto, Japan, (February 2006). ATR Media Information Science Laboratories.

[5] Noriaki Kuwahara, Haruo Noma, Kiyoshi Kogure, Norihiro Hagita, Nobuji Tetsutani, and Hiroshi Iseki, 'Wearable auto-event-recording of medical nursing', in *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2003)*, (September 2003).

[6] Mik Lamming and Mike Flynn, 'Forget-Me-Not: Intimate computing in support of human memory', in *Proceedings of FRIEND21, the 1994 International Symposium on Next Generation Human Interface, Meguro Gajoen, Japan*, Meguro Gajoen, Japan, (1994).

[7] Carolin Plate, Nathalie Basselin, Alexander Kröner, Michael Schneider, Stephan Baldes, Vania Dimitrova, and Anthony Jameson, 'Recomindation: New functions for augmented memories', in *Adaptive hypermedia and adaptive web-based systems: Proceedings of AH 2006*, eds., V. Wade and H. Ashman. Berlin: Springer, (2006).

[8] *The blogosphere*, ed., Andrew Rosenbloom, volume 47 of *Communications of the ACM*, ACM Press, New York, NY, USA, December 2004.

Adaptation of Ubiquitous User-Models

Andreas Lorenz[1], Andreas Zimmermann[1]

Abstract. This paper illustrates the evolution of user-modelling approaches. Starting from the drift of the computing metaphor towards ubiquitous computing, the paper follows the requirements on user-modelling illustrating a parallel drift in its approaches. Furthermore, this contribution identifies the need for adaptation of ubiquitous user-models addressing the specific requirements associated with a ubiquitous computing environment. In order to correspond to this adaptation need we present a taxonomy of adaptation methods that are required for adjusting user-models in a ubiquitous computing environment.

1 INTRODUCTION

In the classic approach for personalized system development, the application contains specific user-modelling components inside the personalization engine. By user model acquisition, information about the user is extracted from sensoring the environment and knowledge from explicit and implicit user feedback is inferred [13]. Traditionally, the inferred knowledge is written to an internal database, mapping user attributes to their values. In the next step, both the component listening to sensor data and the knowledge acquired about the user are separated from the internal application logic. On the one hand, sensor servers retrieve data streams from different sensors placed in the environment and deliver the information to the application. Using remote sensor servers distributing sensor data over a network, different applications can concurrently receive the same data. On the other hand, User-Model Servers [8] work as an application-external knowledge-base. The derived knowledge about the user is delivered to the server that hosts the information for different applications. For mobile applications, this enables systems on small devices even with limited memory and computing power to have access to meaningful user models. Furthermore, it enables different applications to have access to the same knowledge and to adapt consistently.

Conjoint with the evolution of the computing metaphors, the requirements on the knowledge-base changed, no matter if internal or external one. In this paper we will first outline the evolution of the metaphor, currently ending in ubiquitous computing, and will then analyze different approaches of user modelling to fulfil the requirements. Derived from the conclusion of the approaches currently available we will focus on the fulfilment of requirements specifically introduced by ubiquitous computing. Particularly, this means an analysis of specific requirements introduced by ubiquitous computing and how these requirements have influence on the maintenance of a robust, consistent, expressive and effective user-model. In a next step, we elaborate a taxonomy covering potential means for adaptation of a ubiquitous user-model.

2 From the Desktop Metaphor towards Ubiquitous Computing

This section documents the step-wise transition from desktop to ubiquitous computing. Taking the technological aspects as a basis the subsequent section exceeds this view through addressing the evolution of the aspects of user-modelling and adaptation.

2.1 Desktop-Metaphor: One Information Space Connected with one Stationary Device.

In the desktop-metaphor, different information spaces are specifically connected with stationary computers. If one person logs into different computers, she will have access to different information. A change at one personal computer does not affect other computers.

Adaptation processes rely on user-specific information, like preferences and knowledge. Applications store user profiles on the local hard disk. The position, environment and device-attributes are fixed before runtime of the application.

2.2 Mobile Computing: One Information Space per Mobile Device.

Mobile Computing is a generic term describing your ability to use technology to wirelessly connect to and use centrally located information and/or application software through the application of small, portable, and wireless computing and communication devices. In mobile computing the user is not bound to a certain location when using her familiar information device. Like desktop-PC's, different mobile devices have their own information spaces. A change at one device does not affect other devices, neither mobile nor stational ones. To update the different information spaces, explicit synchronising messages between the devices or with servers need to be exchanged.

Besides user-specific adaptation, the system takes changes in the environment into account. There is no need for further adaptation to the device once the user has selected her device at login. Consequently, the system can make use of pre-formatted content.

2.3 Nomadic Computing: One Information Space for all Personal Devices.

Nomadic computing offers its user access to data or information from any device and network while he or she is in state of motion. The nomadic user has just one information space for different personal devices, no matter if mobile or stationary. Changes at one device affect the overall information space and hence other personal devices. There is no need for explicit synchronizing different devices triggered by the user. In addition to the adaptation needs in mobile computing, the nomadic system must be able to identify the current device in use and to adapt dynamically to its characteristics.

[1] Fraunhofer Institute for Applied Information Technology, Schloss Birlinghoven, 53754 Sankt Augustin, Germany, {Andreas.Lorenz, Andreas.Zimmermann}@fit.fraunhofer.de

2.4 Ubiquitous Computing: One Information Space Independent of Devices.

As a fundament of ubiquitous computing, distributed computing is parallel computing using multiple independent computers communicating over a network to accomplish a common objective or task. The vision of ubiquitous computing is a surrounding with the possibilities of gaining information everywhere by many different information devices. Ubiquitous computing integrates computation into the environment, rather than having computers which are distinct objects. In ubiquitous computing, the user has one information space independent of devices. There is no need for personal devices. Consequently, changes affect direct the system and are not device-specific.

In addition to the adaptation needs in nomadic computing, the system manages the personal information space of its users and has to assure that private information is kept confidential. For each single user, her specific task and current situation, the system selects the most appropriate device for interaction. In enhancement of distributed computing, the structure of the service and the structure of human-computer interaction are subjects to adaptation considering e.g. different personal preferences, contexts or user goals.

The provision of personalized information services becomes a complex task in open and distributed environments of mobile users. The aim of distributed systems is to spread the application logic among different parts hosted on different physical devices and hide technological details of the system from the user [20]. This can be a network of standard desktop-PCs where any user has dedicated profiles for each login as well as any client-/server architecture where any user has one central information space on the server-side. Facilitating communication and coordination of distributed components in standard middleware, the complexity introduced through distribution is handled by means of abstraction. Implementation details are hidden from both users and application designers and encapsulated inside the middleware itself [14].

2.5 Summary

In the vision of ubiquitous computing the user has one personal information space independent of devices and the system manages the information spaces of its users. These information devices have a direct contact to each other in order to offer common services. To have access to this personal information space, distributed components on heterogonous platforms need to exchange information with one another. Beside the complexity of techniques to fulfil the classical requirements of adaptive information selection and presentation [2], passing all data from clients to servers for analyzing and centralized decision finding will put both the networks and servers out of business. We therefore think that applications in ubiquitous computing will require distributed user-model acquisition and user-model application. Furthermore, central user-modelling servers holding all information from registered users will not be applicable for two reasons: On the one hand, every device of the mobile users will not have permanent access to the server, and on the other hand, the local system will not constantly need all-embracing knowledge about the user to make a local decision. Each local component might detect a section of the global state, but the network of components must piece together these partial states for distributed representation of knowledge about the user.

In our vision, components for user model acquisition and user model application are equally distributed with the user model itself: Distributed user modelling approaches need to replace monolithic centralized user modelling by distributed user model fragments [19]. As we will describe later, the dynamic adaptation of the structure of the distributed user model will become important to step forward from nomadic towards ubiquitous computing.

3 Overview of User-Modelling Approaches

In the classical adaptive loop introduced by Brusilovsky [1] the system collects data about the user, processes user modelling and adaptation and in turn the effect of the adaptation is recognized as relevant input for the system.

3.1 Information Processing in Adaptive Systems

Based on internal interaction sensors or external environment sensors, all incoming events are collected and reported to a central unit. This central analysing unit handles the incoming event messages, evaluates them and generates decisions on adaptation activities. The result consists of actions executed by corresponding effectors for adaptation. More concrete, Jameson [12] introduced the process of adaptive systems by user-model acquisition, and user-model application, connected by the central user-modelling unit. In this section we illustrate the basic approaches for traditional user-modelling.

Introduced by the step from stationary desktop computing to mobile applications, context-aware systems as an extension of user-adaptive systems exploit the context to increase the fit of a service to the user's needs beyond the user centred evaluation. Context-awareness does not only include the momentary value of variables like location, time, environment, domain, physical conditions and social actors but also their evolution over time. Two technologies allow users to move around with computing power and network resources at hand: portable computers and wireless communications. These technologies enable people to access their personal information, corporate data, and public resources anytime, anywhere with their personal devices. The relationship between systems and devices is constantly changing due to user mobility.

From a general point of view, [7] define context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application*". Focussing on what kind of information describe the subject's context, [17] define *user context, computing context* and *physical context* or [9] refer to *identity*, *location*, *time*, and *environment.* Additionally, when the context-information is recorded across a time span, we can obtain histories like *interaction history, movement history*, or *event histories*, which could also be useful for certain applications [3].

The information delivered by sensors contributes to a knowledge-base of user-related knowledge, and keep its content up to date. This knowledge-base, usually referred to as User Model, will finally form the basis for all decisions taken by a system in order to implement adaptive behaviour in ubiquitous computing. The next sub-sections outline the traditional approaches and discuss their relation to the above-mentioned application areas.

3.2 Monolithic User-Modelling

In the classic approach for personalized system development, the application contains specific user-modelling components inside the personalization engine. This engine is closely connected with the application logic sharing the same host. It is therefore necessary to have local sensors connected with user model acquisition methods, or to be able to connect directly to sensors by the application. The derived knowledge usually is written to an internal database, mapping user attributes to their values.

In the first place, monolithic user-modelling and integrated personalization engines work for all applications that do not share user-attributes with other applications, including applications running on Desktop-PC’s as well as for isolated mobile applications. As an important advantage, the close relation to the internal application logic reduces complexity of system design, user-modelling overhead and network traffic resulting in potential increased performance (memory, time). Nowadays, the installation of a database for persistent storage of user-data does not confront Desktop-PCs with performance problems. In contrast to personal computers, the need of a local knowledge-base is often incompatible with the limited resources of mobile devices.

For sharing user-attributes, e.g. if the user wants to work with the same setup on different devices, applications need to explicitly synchronize with other applications. On the one hand, the synchronization annoys users and introduces extra effort. On the other hand, it introduces network traffic, and potential complex methods to avoid inconsistencies.

3.3 User-Model Server

To enable different applications on different hosts to have access to the same knowledge and to adapt consistently, the knowledge-base about the user could be separated from the internal application logic. User-Model Servers [8] work as an application-external knowledge-base. The inferred knowledge about the user is delivered to the server that hosts the information for different applications. Furthermore, this enables small devices even with limited memory and computing power to have access to meaningful user models.

One domain requiring such servers are web-based services, where users access the application from different browsers. For accessing the information, just a web-browser will be required no matter if the application runs on a desktop or on a mobile device. Since the user-modelling server has usually no write-access to the server-side hardware for writing user-profiles, the server installation comes with its own database. Typically, the extra effort is only reasonable if a large number of users are registered to the service. For single applications on a Desktop-PC, the installation of a user model server is possible but introduces waste of local resources needlessly.

If mobile applications have access to the network, user-model servers have also advantages for the mobile applications as a central access point to the user-model, e.g. for location independent access from any place. As a consequence, the step towards nomadic systems is well-supported by user-model servers providing access from different devices / applications to one and the same knowledge-base. In terms of the discussion in the last chapter, different mobile applications would potentially have different schemas on several servers, or even access completely different user-model servers. Then, for a consistent management of user-models, additional effort will be necessary. If there is a central server with static network-access and constant address available in ubiquitous computing, applications can fall back on a user-model server as knowledge-base. This is often the case if the surrounding services are pieces of the body of one and the same ubiquitous service deployed by a specific vendor.

3.4 Modularization of User Models

The main objective of approaches like the Component-based Architecture of Chepegin et al. (CompAS-project [4][5]) is the modularization of knowledge models allowing sharing user models between applications. The architecture focuses on the role of external user models. To give access to a centralized User Modelling Service, the common user model exchange language UserML [11] supports the communication between different user adaptive systems, which base their decisions on the state of the same central user model.

The modularization enables the interchange of only parts of an overall model: Application A and application B are enabled to put together their information and any application is enabled to download it from the server. If several modules overlap, specific conflict resolving will be necessary. By definition, the overhead of this approach averts its usefulness for the desktop-metaphor. If conflict-resolving is implemented, this approach is expected to especially perform well for nomadic computing where different applications on heterogeneous devices access one and the same personal information space. In turn, the modularization supports downloading of user-model fragments that actually were brought in by another party. For ubiquitous computing, the ability of exchanging separate modules between applications is of important advantage; based on a central user-model server the approach is appropriate if the server is accessible at anytime when adaptation is to be performed.

3.5 Component-Based User Model Framework

Recent approaches (such as [21]) aiming at a modularisation of user-adaptive systems cluster their functionalities in components for sensing, modelling, controlling and actuating. The clusters made up of these components build upon each other [22]. Sensing components collect and record data about the users and their environment in a systematic and continuous way. Based on the data by the sensors, modelling components generate semantically enriched information that serves as a source for subsequent adaptation procedures. Inference methods or rule systems enable the controlling components to interpret this information and take decisions for adaptation. Finally, actuation components handle the connection back to the application by mapping the decisions taken by the control into real world actions in order to realize the selected adaptive behaviour. The actuating function transfers the results of the inferences into respective options of operations adapting the functions or the interface of the system to the current user's needs.

First, this approach is a good candidate for application development on both desktop and mobile devices: the framework specifically supports system development and integration of sensors, user-modelling, and actuators; whereas at runtime the overhead is limited. By modular system design and specific interfaces between components of different functionality, the increased scalability of the framework also supports the development of tailored mobile applications. For nomadic and ubiquitous computing, the advantages are comparable with the above-mentioned for modularized user models. Whereas, the most important advantage for ubiquitous computing is the support of distributed components on each single layer, which specifically benefits integration of surrounding sensors/actuators in ubiquitous computing.

3.6 Distributed User Modelling

For the mobile user, her device will have partially access to the knowledge. The devices will continuously connect to local networks and therefore will have access to all information available in this network. A centralized solution fails because of its inability to cope with a high degree of change, which requires that the solution is both robust to disruption end self-configurable. For distributed user modelling approaches this implies that monolithic user modelling is replaced by distributed user model fragments [19]. The knowledge about the user, i.e. the current state of the user-model, will be merged from all information that can be requested from components reachable in the current context. Even if for example a mobile device like a PDA may host different sensors like for noise or light and may implement several display components and actuators for video-streaming and adjustment of the screen backlighting, a server is still needed for modelling the users and for controlling the application.

Using remote sensor- and actuator-servers distributing data over a network, different applications can concurrently receive and send data. Resent research in smart sensor-networks enables for placing huge numbers of intelligent senor-components ("smart dust") in the environment. Smart sensors are equipped with small processors that enable for intelligent information acquisition [16]. In self-organizing networks, such as Intel's iMote approach [6], sensor technologies build ad-hoc sensor-networks and deliver requested information on demand. In difference to such sensor-networks, distributed modelling components actually receive pre-processed data from virtual components instead of direct measuring the physical environment.

3.7 Discussion

The entire adaptation process bases on a modelling layer that supplies the controlling components with an accurate image of the current interaction situation. The context model captures the current situation the users act in, their preferences, interests, their social dependencies, their physical and technical environment, and so forth. Overall, there exist many different views on what dimensions such a model has to cover, e.g. Identity, location, time, and environment [9] or user context, computing context and physical context in [17] and types of contexts, like primary and secondary contexts [7] or static and dynamic contexts [21].

Given the complexity of such a model it can be seen that the sources for acquiring information for this model and means to realize adaptation needs based on this model can be manifold. In particular, in mobile applications or even more in ubiquitous environments these two aspects, sensing and actuating, may be inheritably distributed. Such applications rely on a network of sensors placed within the physical environment and watching indicators for changing situations. On the other hand, the actuators are specialized software-components that process the delivered data or display information snippets on a particular device. Depending on the integration with the application, sensing and actuating components may be part of the target application.

If a user-adaptive application for ubiquitous computing completely lost connectivity, the application has only access to local information. In case of a central remote user model, the application will not have any knowledge, because of the unreachable server. Only denoting the other extreme, i.e. if the system has full access to the server, it can rely on a global user-model accessing all knowledge. Due to constant changing user-environments in ubiquitous computing we expect the diffuse state in between to be the most relevant one.

4 Ubiquitous User-Modelling

Providing a framework will enable all applications on the devices to check into the network and to make use of the available user-related information. Usually, this information will just be a cut-out of all potentially known knowledge, except in case of every existing information-source is accessible. Nevertheless, the cut-out will reflect the current environment by including all relevant information at the current location automatically. In the next section we will illustrate our approach for supporting the information-exchange between ad-hoc networked components.

4.1 Definition of Ubiquitous User-Modelling

Embedding computation into the environment and everyday objects would enable people to move around and interact with information and computing more naturally and casually than they currently do. One of the goals of ubiquitous computing is to enable devices to sense changes in their environment and to automatically adapt and act based on these changes based on user needs and preferences.

Using the term *ubiquitous user-modelling* equally with the term *user-modelling **for** ubiquitous computing*, many approaches apply existent user-modelling approaches to meet the specific requirements of ubiquitous computing (e.g. [14]). Especially approaches falling back on central user-model servers for sharing (parts of) user-models performed very well under the condition that the server is accessible at the time adaptation is performed. From the changes of the computing metaphor we know, that this might not be true for ubiquitous services. The vision of ubiquitous computing is going beyond distributed mobile computing.

Getting back to the vision [20] that "you don't bring the computer with you, it is already there" we conclude by analogy that system components bring user-model parts with them and other parts are already there. As [1] pointed out, *"the structure of the clusters is highly dynamic, as they comprise user's devices (providing partial user models), whose availability is unstable."*, whereas [10] pays specifically attention to the requirement of sharing user models: *"Ubiquitous user-modeling describes ongoing modeling and exploitation of user behaviour with a variety of systems that share their user-models."*

Like the structure of a ubiquitous service is subject of change depending on the environmental settings and personal attributes, the structure of the user-model itself is also subject of adaptation. From this point of view, a ***Ubiquitous User-Model*** *is a base of knowledge that is currently available about a user of a ubiquitous service which structure is subject to change unpredictable and without explicit notificatio*n. Consequently, the process of user-modelling must consider that the composition of the user-model is only valid in a specific situation: ***Ubiquitous User-Modelling*** *describes the exploration, representation and exploitation of that knowledge inside system components, the sharing including conflict-resolution of user-model fragments between system components, and the adaptation of the ubiquitous user-model itself to the environmental settings.*

4.2 The Need for Adaptation

Since a potential user-modelling framework operates in a ubiquitous computing environment, this framework as well as the user-model needs to address respective requirements on ubiquitous computing. Taking into account physically distributed real-world entities it becomes apparent that distribution plays a significant role. Distribution occurs on different levels: on a conceptual level where *information is distributed* (e.g. fragmented and distributed user-model) and on an implementation level where *system components are distributed* [18]. The architecture should support both distributed and co-hosted implementations.

The heterogeneity of devices, programming languages and operating systems exploited by a context-aware application in a ubiquitous computing environment is immense. This includes aspects like the format of the data they provide, their interfaces, mode of operation, coverage, and limitations as well as economic criteria such as costs, etc. An integration of such arbitrary components into a coherent infrastructure requires a diligent consideration of their heterogeneous characteristics and capabilities.

A ubiquitous computing application need to cope with many forms of spontaneity. Services may fluctuate, devices may disconnect and the quality of a service may decrease. The parts of the user-model provided by these services are subject to a dynamic environment and needs to deal with component malfunction, jam, disappearing, restart, or inappropriate output. Facing these ubiquitous computing challenges a ubiquitous user-model needs to be maintained and adapted to dynamic changes.

4.3 Adaptation Means for Ubiquitous User-Models

The preceding section identified the need for the continuous adaptation of ubiquitous user-models. The goal persuaded by an initiated adaptation process is the retention of a consistent, efficient, effective and robust user-model. Sources for adaptation comprise the domain model, the context-model and the user-model itself. Independently from the adaptation target, i.e. the ubiquitous user-model, an array of basic means and methods for adaptation can be specified.

An overview of several basic adaptation approaches is given in Figure 1. There exists an increasing complexity from null adaptation on the left, to generative approaches on the right hand side.

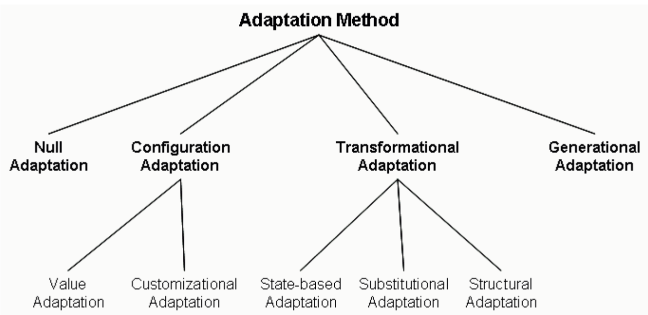

Figure 1 Hierarchy of Adaptation Methods

The following sections will describe these different approaches, provide an overview on each adaptation method, and describe the knowledge required to perform this adaptation. For each method, an short example will be given.

4.4 Null Adaptation

The simplest way of adaptation is the null adaptation, if the system recognizes that in the current situation adaptation is not necessary or of no benefit. Null adaptation can also mean that the adaptation of the user-model is entirely left to the user.

4.5 Configurational Adaptation

The term configurational adaptation refers to changes affecting the internals of elements or components of the ubiquitous user-model. This adaptation changes accessible values of these components and alters parameters, in order to reconfigure their behaviour. The configurational adaptation can be subdivided into value and customizational adaptation.

4.5.1 Value Adaptation

A basic way of adaptation exhibits the value adaptation as the first form of configurational adaptation. Value adaptation specifically affects elements of the ubiquitous user-model that serve as containers for values. This adaptation method addresses the alternation of values contained by these components. The functionality of the concerned component remains constant, as well as the structure of the ubiquitous user-model. The knowledge necessary for this adaptation mean consists of possible ranges for values and their type. An example for a value adaptation is the continuation of the location determination through extrapolation (e.g. history analysis), during signal loss of a GPS device in a tunnel.

4.3.2 Customizational Adaptation

Customizational adaptation influences the functional behaviour of elements of the ubiquitous user-model. Selective alternations of specific parameters of a component lead to changes in its behaviour or view. Independently developed adaptive applications are constructed under a set of assumptions that the developer had to make about the target operating environment. Functional configuration allows for a behaviour extension without the need for re-implementation. Customization and reconfiguration are the prerequisites for the universal applicability of components. Customizational adaptation does not affect the interfaces to other components. The knowledge required for this kind of adaptation has to cover the functionality of affected components and means for their parameterization. The reduction of the event triggering rate of a tracking component to cope with a low bandwidth is indicative of customizational adaptation

4.6 Transformational Adaptation

Transformational adaptation means that the old structure or composition of the ubiquitous user-model is transformed into a new one. This kind of adaptation supports the reorganization of parts of the ubiquitous user-model and permits modification, addition and removal of these elements under certain conditions. Typically, systems performing transformational adaptation employ a fixed set of adaptation operators or transformation rules. Transformational adaptation requires domain knowledge on how certain changes in the structure of the ubiquitous user-model lead to differences in its behaviour. Depending on the degree of modification state-based, substitutional and structural adaptation can be distinguished.

4.4.1 State-Based Adaptation

The state-based adaptation method covers all changes to the states of components. Possible switches between component states correspond to activating or deactivating components. As a result of this adaptation method, the (de-)activated component is still present in the structure of the ubiquitous user-model, and thus, its structure remains unmodified. State-based adaptation complies with a controlled intervention into the structure of the ubiquitous user-model. The application of this adaptation method requires knowledge about dependencies of the affected component with other components within the structure. An example of state-based adaptation is the deactivation of a component because of a sensor malfunction.

4.4.2 Substitutional Adaptation

The substitutional adaptation is related with the state-based adaptation method. It addresses the replacement of one element of the ubiquitous user-model with another. The result of adaptation method will typically be very close to the initial situation. The structure of the ubiquitous user-model remains unchanged. This adaptation method requires similar components to be replaced with each other depicting an approximately similar behaviour. Additionally, the exchanged components should provide the same interfaces for accessing their functionality. Taking the determination of the user's location as an example, a substitutional adaptation occurs if two tracking technologies like GPS- and WLAN-tracking are exchanged, since one of them delivers a low quality of service.

4.4.3 Structural Adaptation

More substantial modifications to the ubiquitous user-model are performed during the structural adaptation method. Structural adaptation supports the reorganization of elements of the ubiquitous user-model and permits the addition and removal of complete elements under certain conditions. The utilization of this adaptation method employs a fixed set of adaptation operators or transformation rules, which modify the structure of the ubiquitous user-model, depending on the relation between the initial situation and the desired situation. An example for a structural adaptation of a ubiquitous user-model is the dynamic extension of the model with an attribute for the user's location, after the GPS device finished its satellite discovery and starts tracking.

4.7 Generative Adaptation

Generative adaptation is radically different from transformational adaptation. This adaptation method requires a generative (re-) programming from scratch or change of the functionality of an element of the ubiquitous user-model. Such generators need to be tightly integrated with the adaptive system and might perform a generation automatically or in correspondence with the user. In practice a pure automatic generative approach is mostly insufficient, because of the computational complexity of the generation process or because of the insufficient quality of the results it produces. Such an automatic generator should only generate those small parts of the ubiquitous user-model that are inadequate regarding the desired situation.

As a consequence the use of a generative adaptation method requires a different kind of knowledge than for the transformational adaptation. Instead of exploiting knowledge describing how changes of the current situation leads to differences in a potential desired situation, knowledge for constructing (parts of) a solution from scratch is required. For example the adaptation of an inference algorithm to cope with symbolic attribute values could be one instance of a generative adaptation.

5 CONCLUSION

Driven by the paradigm-shift from the personal PC to ubiquitous computing we realized a parallel shift in the personal electronic information space of computer users. In summary, both shifts introduced challenges in personalisation and adaptation. In this paper we illustrated how traditional user-modelling approaches keep up with the new challenges. Under specific circumstances, especially if a central server is available hosting user models (or user-model fragments), traditional approaches are able to cope with the requirements of distributed mobile computing. User-modelling in a ubiquitous computing environment demands the addressing of specific requirements and conditions. Therefore, we presented a set of abstract adaptation methods to be applied for the maintenance of ubiquitous user-models. Particularly, the need for structural adaptation of the user model at runtime distinguishes ubiquitous user modelling from other approaches outlined in the paper.

REFERENCES

[1] S .Berkovsky, `Ubiquitous User Modeling in Recommender Systems.', in L. Ardissono, P. Brna and A. Mitrovic, eds., Proceedings of the 10th international conference on user modelling, pp. 496-498 (2005)

[2] B. Brusilovsky, `Methods and techniques of adaptive hypermedia', User Modeling and User-Adapted Interaction, 6(2-3), 87-129, (1996).

[3] G. Chen and D. Kotz, `A survey of context-aware mobile computing research', Technical Report TR2000-381, Computer Science Department, Dartmouth College, Hanover, New Hampshire, (2000).

[4] V. Chepegin, L. Aroyo, P. De Bra, and D. Heckmann, `User modelling for modular adaptive hypermedia', in Proceesings of the Workshop on Semantic Web for E-Learning, pp. 366-371, Eindhoven, The Netherlands, (2004).

[5] V. Chepegin, L. Aroyo, P. De Bra, and G. Houben, `Chime: adaptive framework for semantic web service integration', in Proceddings of the InfWet-03 Conference, pp. 29-36, (2003).

[6] H. Culler, D.E. and Mulder, `Smart sensors to network the world', Scientic American, 290(6), 52-59, (2004).

[7] A.K. Dey and G.D. Abowd, `Towards a better understanding of context and context-awareness', Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, (1999).

[8] J. Fink and A. Kobsa, `A review and analysis of commercial user modeling serevers for personalization on the world wide web', User Modeling and User-Adapted Interaction, 10(2-3), 209-249, (2000).

[9] T. Gross and M. Specht, `Awareness in context-aware information systems', in Mensch und Computer - 1. Fachuebergreifende Konferenz, Oberquelle, Oppermann, and Krause, eds., pp. 173□182, Bad Honnef, Germany, Teubner-Verlag (2001)

[10] D. Heckmann, `Ubiquitous User Modeling', phd-thesis, Department of Computer Science, Saarland University (2005)

[11] D. Heckmann and A. Krueger, `User modelling markup language (userml) for ubiquitous computing', in P. Brusilowsky, A. Corbett, and F. de Rosis, eds., Proceedings of the 9th international conference on user modelling, pp. 403-407, Johnstown, PA, USA, (2003). Springer-Verlag.

[12] A. Jameson, `Modelling both the context and the user', Personal Technologies, 5(1), 29-33, (2001).

[13] A. Jameson, `Systems that adapt to their users: An integrative overview', in Tutorial presented at 9th International Conference on User Modelling, Johnstown, PA, USA, (2003).

[14] J. Kay, B. Kummerfeld and P. Lauder, `Managing private user models and shared personas', Workshop on User Modelling for Ubiquitous Computting, pp. 1-11 (2003).

[15] C. Mascolo, L. Capra, and W. Emmerich, `Middleware for mobile computing (a survey)', in E. Gregori, G. Anastasi, and S. Basagni, eds., Advanced Lectures in Networking, Springer-Verlag, (2002).

[16] M. Satyanarayanan, `Of smart dust and brilliant rocks', Pervasive Computing, 2(4), 2-4, (2003).

[17] B.N. Schilit, N.I. Adams, and R. Want, `Context-aware computing applications', in Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 85-90, Santa Cruz, CA, (1994).

[18] A.Schmidt, `Ubiquitous Computing in Context', Ph.D. Thesis, Lancaster University, Lancaster, UK (2002).

[19] J. Vassileva, `Distributed user modelling for universal information access', in Universal Access in Human - Computer Interaction, ed., C. Stephanidis, 122-126, Lawrence Erlbaum, Mahwah, N.J., (2001).

[20] M. Weiser, `The world is not a desktop', Interactions, 7-8, (1994).

[21] A. Zimmermann, A. Lorenz, and M. Specht, `Applications of a Context-Management System'. Fifth International and Interdisciplinary Conference on Modeling and Using Context, Paris, France (2005)

[22] A. Zimmermann, A. Lorenz, and M. Specht, `Personalization and context management'. User Modeling and User Adaptive Interaction, Special Issue on User Modeling in Ubiquitous Computing, 15:275–302 (2005).

An Agent-Based Approach Supporting Personal Ubiquitous Interaction

Francesca Muci[1], Pawel Drozda[1], Giovanni Cozzolongo[1]

Abstract. In this paper, we describe an agent-based user modelling approach for personalizing the interaction in ubiquitous and pervasive contexts. To this aim the agent has to model the user behaviour and transfer to the environment what can be inferred about the user in that particular context. Obviously, to achieve this aim, the Personal User Modeling Agent (PUMA) and the Environment have to dialog in a way that they can understand each other. For this reason we propose a schema based on ontology and ontology mapping to solve this human-centered service coordination issue.

1 INTRODUCTION

As far as personalization in ubiquitous and pervasive context is concerned a challenging issue regards the problem of designing the interaction between the user and services in such a way that these can transparently access user preferences and interests. User-related information then, can be used to provide specifically tailored services [1,2].

If we consider the ubiquitous user modelling task, that is fundamental for personalization purposes, this can be performed, in our opinion, in several ways: i) personalization is performed server side, as in the classic adaptive systems [3]; ii) the user has the control over his/her user model that runs, for instance, on a personal device [4]; iii) mixing together the two previous approaches. . This is the approach that we consider in this research. In this way the interaction may become personal even in a public space. In this paper we describe a system in which user-related information may be transferred to the environment and, then, requested information may be adapted to that user; the interaction can be handled through her/his personal device [5], using a Personal User Modeling Agent (PUMA) that manages the user model from the user side [6,7]

In particular, we focus on the coordination problem that arise when this agent has to dialog with smart environments or ubiquitous services that may use a different conceptualization for describing the same domain objects, interests, preferences, etc[8].

Representing user model according to semantic formalisms [9] and sharing ontologies in pervasive, ubiquitous environments enables to share information that can used by many different services [10]. This approach gives a great opportunities in the user modeling field; for instance, different adaptive systems can exploit the same user model without creating their own one.

A common approach to solve this problem is to employ software agents that can exploit information from the ontology in order to create personalised services such as tutorial, recommendation and search services [11,12]. To achieve this aim, the PUMA has been integrated in a coordination schema with a set of agents, based on the semantic web vision, able to gather, process and represent all information about the user referring to ontologies.

In order to describe how this coordination schema works, we developed a running prototype in the context of tourist services. This has been achieved by integrating the PUMA with an existing system called MyMap [13].Furthermore we have developed and implemented an ontology for the tourist domain and for each service provided by the system. MyMap's main goal is to provide help for users in this domain such as finding a place to eat, finding an appropriate hotel, getting information about bus courses, etc.. All services provide personalised suggestions for topics interesting for the user, taking advantage of information gathered from user models, through the use of different ontologies.

The paper is structured as follows: Section 2 presents related work in the field of user modelling connected with Semantic Web technologies. In the Section 3 an overview of the system is briefly described. Finally conclusions and future directions are made in Section 4.

2 RELATED WORK

Recently, computing system specialized in handling real world information in terms of physical objects and relations between them, such as ubiquitous computing systems [14], have received much interest in many research fields. This is due to the complexity of such systems since they integrate in an ubiquitous computing environment devices, services that use those devices and users who have different goals and preferences in interacting with the system.

In such a situation, coordination is important but, in particular is very important to maximize the user's satisfaction. This is possible by trying to approach the problem from a user-centred point of view. Coupling "Semantic Service Agents", which are autonomous computer programs that can use meta-data to understand content in order to provide the appropriate service, and "Personal Agent", which are autonomous agents that know their users and act on their behalf, can be a solution to help users in achieving their intentions when interacting in this complex distributed "world".

Many researches have shown that the Semantic Web presents technologies that allow to solve human-centred coordination issue in ubiquitous computing [15]. There have been proposed many analysis which concern the implementation of the Semantic Web technologies in user modeling systems. Most papers describe researches in e-learning, recommender systems, mobile and ubiquitous computing domains. The problem of creating the domain ontology is presented in [16]. In the paper the authors proposed the collection of ontologies (COBRA-ONT) for supporting pervasive context aware systems. Ontologies encoded in Web Ontology Language (OWL) describe all concepts, properties and relationships between them in an intelligent meeting room domain. In our case there is a need of ontology for describing the tourist and services domain.

The problem of representing user profiles using Semantic Web technologies is addressed in [17] and [18]. In the first are described the PAPI and the IMS LIP standards. PAPI distinguishes personal, relations, security, preference, performance and portfolio information, while IMS LIP contains the identification, goal, QCL, activity, interests, relationship, competency, accessibility, transcript, affiliation and security key. We decided to use UbisWorld language presented in [18] for creating user model ontology. We chose this representation because it allows managing user models

[1] Intelligent Interfaces, Department of Computer Science, University of Bari. Email: {muci,pawel,cozzolongo}@di.uniba.it

in a dynamic way. The language allows also to represent all concepts related to the user, structuring them in situational statements that take the context into account.

Some researchers deal with user modeling in tourist domain. In [19] the construction and management of user profiles for an agent-based travel support system is proposed. As a representation language authors chose RDF, one of the Semantic Web language. The system is designed to be accessible from all the devices with an access to the Internet.

The difference appears in the way of getting personalized information about services. Authors of [19] collect data from Web using concept of VCP (Verified Content Providers) while in our case all information are provided by Service Ontology.

Another example of an ontology in a user modeling system in tourist domain is described in [20]. There are proposed several approaches to realizing adaptive mobile GI services in the domain of pedestrian navigation and tourist information. In another application for tourists [21] the concepts of mobile computing is connected to the one of natural language processing. The main difference to our system is the way of communication with the user: authors of [21] use speech to connect user with the system taking advantage of natural language processing while we use text messages. As repositories of knowledge databases are used in [21] which constrains area of usage of the system when we maintain knowledge in ontologies.

3 OVERVIEW OF THE SYSTEM

To test our hypothesis we have developed a system in which many components collaborate to achieve the user goals. In particular we have realized a running prototype specialized for the tourist domain. Let's suppose that a person named Bob equipped with mobile device is located in one of the part of City of Bari for business purposes and he wants to find a nearby place to eat a dinner. He likes trying some typical dishes, likes a red wine, adores meat and doesn't like chocolate. His preferred cake is "shockcake" and as a starter "Mozzarella with tomatoes and basil". Our system taking into account the user's preferences will try to find out the restaurant that best fits them and will make a reservation of a table for him.

To perform this task, our system includes the main component application: MyMap, a Personal User Model Agent (PUMA), many Service Agents (SA), an Mediator Agent (MA) and a set of specialized Ontology Service, belonging to each service in the system.

The basic architecture and coordination schema of the system described in this paper is shown in Figure 1.

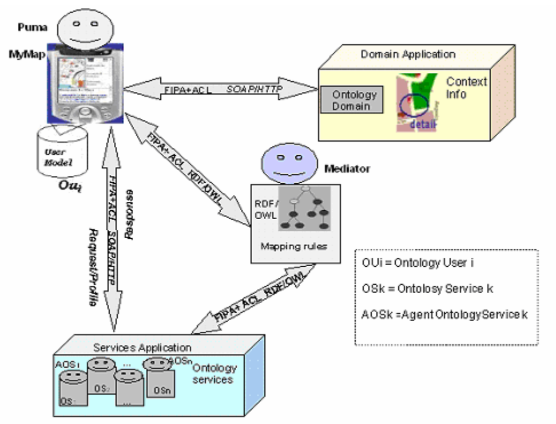

Figure 1. The architecture and coordination schema.

All agent shown in the above figure communicates using ACL messages.

In the next subsections, we describe in detail, each main component of the system.

3.1 MyMap

MyMap is a system developed in our department [22] to provide personalized information to tourists. The map is filled with information filtered on the basis of the user profiles. Places like restaurants, fast foods, hotels, museums, theme parks, etc. are marked on the screen and listed in order to fulfill user needs.

Figure 2 presents a example of interface of MyMap Application. When a user is in an area, the map with a set of point of interest (POI) is shown and a list of items is presented, ordered on the basis of the user preferences. To obtain this information MyMap is connected to an environment, where there are a set service agents responsible for the information about every POI.

Figure 2. An example of MyMap application.

Each element shown in the map is annotated with metadata This annotation is represented in XML and follows the definitions of a Domain Ontology, which express names, terms and attributes that specifies features and their relations between each other.

In our case the domain ontology consists of all terms connected with tourism.

All components are organized in a hierarchy of classes and relations between classes encoded in OWL [23].

A part of structure of Domain ontology regarding the services Restaurant and Cinema is listed in Table 1:

Table1: A portion of the Domain Ontology

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/mymap.owl#"
 xml:base="http://www.owl-ontologies.com/mymap.owl">
 <owl:Ontology rdf:about=""/>
…
<owl:Class rdf:ID="Restaurants"/>
 <owl:Class rdf:ID="Restaurants_for_celiac">
  <rdfs:subClassOf rdf:resource="#Restaurants"/>
 </owl:Class>
 <owl:Class rdf:ID="Restaurants_tipical">
  <rdfs:subClassOf rdf:resource="#Restaurants"/>
 </owl:Class>
 <owl:Class rdf:ID="Restaurants_chinese">
  <rdfs:subClassOf rdf:resource="#Restaurants"/>
 </owl:Class>
```

```
 <Restaurants_tipical rdf:ID="Ristorante_Le_mura"/>
   <hasLocation rdf:resource="#mapZone4"/>
    <rdfs:comment
       rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Nice
       rustic tone, in this central restaurant, with a wood stove in the eve-
       ning pizzeria and traditional dishes; pleasant outside in sum-
       mer</rdfs:comment>
       <hasName
       rdf:datatype="http://www.w3.org/2001/XMLSchema#string">leMur
       a</hasName></Restaurants_tipical>
       <Restaurants_tipical rdf:ID="Ristorante_da_Federico"/>
       …
   <owl:disjointWith>
    <owl:Class rdf:ID="Cinema"/>
…
</rdf:RDF>
```

3.2 PUMA

To manage the preference of the user MyMap is connected to a Personal User Model Agent (PUMA). This agent is specialized in user modeling features. Furthermore it is responsible for managing the to-do-list of the user, so as to know his agenda. When a task is activated the PUMA will try to achieve it exploiting all the services provided by the environment.

If the task is that of reaching a place the PUMA contacts MyMap. To find out what is the best place to reach the PUMA exchanges some information about the user with the services related to a given place. Then it chooses the one that best fits the preferences and informs the user, which can agree or disagree. In the case of our example, the PUMA will look for a restaurant and will chose a “Typical Italian Restaurant”, whose menu contains the first plate “rice, potatoes and mussels”, a typical dish of Bari. The PUMA will go on performing the order, until is able to understand all concept in the received menu.

However, the information inferred by the PUMA has to be passed to the environment in a “understandable” way. Instead of defining a new ontology and language for describing mobile user profiles, since this is not the main aim of our research, we decided to adopt UbisWorld [24] language.

In this way we have a unified language able to integrate user features and data with situational statements and privacy settings that better suited our need of supporting situated interaction. This language allows representing all concepts related to the user by mean of the UserOL ontology, to annotate these concepts with situational statements that may be transferred to an environment only if the owner user allows this according to privacy settings. A situational statement regarding the user modelling preferences can be represented as in Table 2.

The ontology is a collection of user profiles concepts. User model is composed of data which are necessary to provide essential information about user for the system.

These data include demographic information (like sex, age), permanent preferences and interests (for example interested in soccer, vegetarian, amount of money ready to spend for dinner, favorite dishes), temporary preferences and interests dependent of context (for example I like eating typical food in other country, I like eating outside when sun is shining), user situation (available time for dinner, level of hunger).

The PUMA manages and observes the user actions in the whole interaction. When new information about him can be inferred, it updates the user profile by adding the new information. It contacts the SAs, of the service needed by the user, sending the portion of user profile with preferences about the requested service.

Table2. A Portion of user profile

<situation> <statement id="1"> <mainpart subject="Bob" auxil- iary="HasEatingPreference" predicate="Dessert" range="Text" object="Shockcake" /> <constraints start="2006-04-12 12:27:08" duration="Year" /> <explanation creator="Bob" confidence="0.75" /> <privacy owner="Bob" access="Public" purpose="Eating"/> </statement>… </situation>	<?xml version="1.0"?> … <owl:Class rdf:ID="Eating_Preference"> <rdfs:subClassOf> <owl:Class rdf:ID="User_Prefence"/> </rdfs:subClassOf> </owl:Class> …. <owl:Class rdf:ID="Dessert"> <rdfs:subClassOf rdf:resource="#User_Preference"/> … <Dessert rdf:ID="Shockcake"/> …. </rdf:RDF>

Then, when moving in a mobile or ubiquitous environment, the spreading of user profile fragments is natural; users are controlled by different devices and the user information collected by these devices typically makes sense only in the context (e.g. for a short period of time, for a given place etc.).

3.3 Service Agents

Service agents, in this case, are specialized in providing information about places of interest, such as a restaurant, a bus station and so on. They provide the type of service/place they manage, and other features useful to better select them for the user.

For each service provided in tourist domain, a separate ontology is constructed. Each of them contains structure of concepts and relations between them specific for that service. In addition they are provided with a part of users’ profiles which are interesting for each services. For example the restaurant ontology will have all dishes proposed in menu with prices and ingredients, opening hours, number of available tables, average time spent in restaurant etc. Ontology of bus company contains time tables, list of bus stops, routes of buses, times needed to get from one place to another. A part of Restaurant Ontology is shown in Table 3:

Table3: A portion of ontology service

```
<?xml version="1.0"?>
<rdf:RDF
…
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Ristorante_Le_mura"/>
 <owl:Class rdf:ID="Desserts">
   <rdfs:subClassOf rdf:resource="#Ristorante_Le_mura"/>
 </owl:Class>
 <owl:Class rdf:ID="Second_plates">
   <rdfs:subClassOf rdf:resource="#Ristorante_Le_mura"/>
 </owl:Class>
 <owl:Class rdf:ID="First_plates">
   <rdfs:subClassOf rdf:resource="#Ristorante_Le_mura"/>
 </owl:Class>
….
 <Second_plates rdf:ID="horse_meat"/>
….
 <First_plates rdf:ID="Rice_Potatoes_mussels"/>
….
 <Desserts rdf:ID="Strawberry_cake"/>
…
</rdf:RDF>
```

In our example the Restaurant Service Agent suggests that in menu there are starters like ”Set of cheeses”, “Hot vegetables” and “Smoked meats homemade”, second plates like “sausages” and “shrimps”, dessert “tiramisu” and “fruit salad” and typical regional red wine “Castel del Monte” based on Bob’s preferences.

3.4 Mediator Agent

The Mediator is an agent which mediates between the different terminologies or conceptualization in the individual data sources and helps the communication of other agents. With the term “different terminologies” we mean the same term may have different meanings (polysemy), or the same meaning may be associated with different term and/or representations (synonymy).

Several different tools are available, that include techniques for mapping, integrating and merging ontologies, sub-ontology factoring, but there is no automatic way to do that. It is still a difficult task and for the success of these processes it is necessary to detect ontology mismatches and solve them. One way to do this mapping is to use the “most similar” concept in the first ontology and find them in the second ontology [10, 12].

The solution we have adopted, at the design level, is to provide the MA with the autonomy for word sense disambiguation. The MA could be invocated either by the PUMA or by the SA, depending on which of them is not able to understand a term. To help the other agents to complete this task the MA, first applies the algorithm of similarity. It asks the PUMA and SA the two graph-like structure of the ontologies and analyses them using a parser to find name classes, property or instance in SAO or UOi which corresponds to dubious term. If it does not complete the task of disambiguation term, MA negotiates with PUMA (if it was invocated from SA) or with SA (if it was invocated from PUMA) asking for additional information. Then they exchange messages which help to resolve conflicts of similarity.

In our example, the problem raises when choosing a dessert. The personal agent couldn’t find anything similar to Bob’s favourites fruit cake with strawberry called “shockcake”. In this case the personal agent and restaurant service agent have to agree what “shockcake” means. To achieve this goal they contact the Mediator Agent. The PUMA sends to the MA which type of sweets the “shockcake” is (ice creams, cakes, etc.). Then the restaurant agent is asked to send the list of type of cakes. The PUMA selects “fruit cakes” and the restaurant service agent lists all fruit cakes that are available. The mediator then map the “shockcake” as a “cake with strawberries”.

4 CONCLUSION AND FUTURE WORKS

In this paper, we have described the prototype of a system able to generate context-sensitive description of objects in a map for supporting tourists. Even if we selected the mobile tourism as a application domain to test our approach, the system architecture and employed methods are general enough to be applied to other domains.

In order to describe how this interaction schema works, we developed a running prototype in the context of tourist services. This has been achieved by integrating the PUMA with an existing system called MyMap.

We are using JADE agent environment for the PUMA and OWL to create the ontologies. We have chosen OWL since it has been designed as a standard and as the baking of a well known and regarded standard organization W3C. For these reason, there is a wide variety of development tools available for integrating ontologies into the development of our software application. Between this tools, we have chosen Protégé 3.1 [25] for creating and mapping of ontologies.

At this point of our research we use directly users’ features, interests and needs in order to provide appropriate suggestions. In the future when the system will be used by many users we want to adopt collaborative filtering [26] for creating stereotypes of users in all services. Each service will assign any new user of the system to one of the stereotypes created from interaction of previous users and will give suggestions on the basis of that stereotype. The stereotype will be built starting from the profiles of users which have similar characteristics. The similarity will be measured using an algorithm based on distance vector [27].

Acknoledgements

We would like to thank Prof. S. Pizzutilo and Prof.ssa B. Decarolis for their support and helpful suggestion in developing our research.

REFERENCES

[1] De Carolis B., Pizzutilo S. and Palmisano I. D-Me: Personal interaction in Smart Environments. In P Brusilovsky, A Corbett and F de Rosis (Eds): Proceedings of User Modeling '03.Springer LNAI 2702. 2003

[2] Dey A. K. Understanding and Using Context. Personal and Ubiquitous Computing 5 (2001) 1, 4-7.

[3] Brusilovsky P.(1996).Methods and Techniques of Adaptive Hypermedia.UMUAI,6.87:129.

[4] Cozzolongo, G., De Carolis, B., Pizzutilo, S.. Supporting Personalized Interaction in Public Spaces. In Proceedings of the Artificial Intelligence in Mobile Systems 2004. Baus J., Kray, C., Porzel, R. (Eds.). Nottingham, UK, 2004.

[5] Cavalluzzi A., De Carolis B., Pizzutilo S., Cozzolongo G.: Interacting with embodied agents in public environments. AVI 2004: 240-243.

[6] Kobsa A., Generic User Modeling Systems. UMUAI vol. II nos.1-2 pp.49-63. Kluwer Academic Publisher. 2001.

[7] Fink J. and Kobsa A. (2002): User Modeling in Personalized City Tours. Artificial Intelligence Review 18(1), 33-74.

[8] Bickmore T. and Cassell J. Relational Agents: A Model and Implementation of Building User Trust. ACM CHI 2001 Conference Proceedings, Seattle, Washington, 2001.

[9] F. Carmagnola, F. Cena, C. Gena, I. Torre, A Multidimensional Approach for the Semantic Representation of Taxonomies and Rules in daptiveHypermedia Systems. Dipartimento di Informatica, Università di Torino, 2005.

[10] Christopoulou E., Kameas A., “GAS Ontology: an ontology for collaboration among ubiquitous computing devices”, International Journal of Human – Computer Studies: special issue on Protégé, (in press) 2004.

[11] Konstan J.A. and Riedl J. Research Resources for Recommender Systems. CHI' 99 Workshop Interacting with Recommender. Pittsburgh, Pennsylvania, USA 1999.

[12] Akio Sashima, Noriaki Izumi, Koichi Kurumatani Bridging Coordination Gaps between Devices, Services, and Humans in Ubiquitous Computing. Workshop on Agents for Ubiquitous Computing. 2004.

[13] B. De Carolis, G. Cozzolongo and S. Pizzutilo: Generating Personalized Tourist Map Descriptions, In Proceedings of IEA/AIE 2005: pagg. 259-268 , Jun, 2005.Bari, Italy.

[14] Weiser, M.: The Computer for the 21st Century. Scientific American, 94–104, 1991.

[15] Berners-Lee, T., Hendler, J. and Lassila, O., the Semantic Web. Scientific American, 2001.

[16] H. Chen, T. Finin, A. Joshi. An Ontology for Context-Aware Pervasive Computing Environments, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, Page 197—207, No. 3, Vol. 18, Cambridge University Press, May 2004.

[17] P. Dolog, W. Nejdl. Challenges and Benefits of the Semantic Web for User Modelling. Proceedings of the Adaptive Hypermedia Workshop at WWW 2003.

[18] D. Heckmann. Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource-adaptive computing. In: ABIS2003, Karlsruhe, Germany (2003) 283–286.
[19] Gawinecki, M., Gordon, M., Paprzycki, M., Vetulani, Z.: Representing Users in a Travel Support System. In et. al., H.K., ed.: Proceedings of the ISDA 2005 Conference, Los Alamitos, CA, USA (2005) 393–398.
[20] A. Zipf, M. Jöst. Implementing Adaptive Mobile GI Services based on Ontologies - Examples for pedestrian navigation support. In: CEUS - Computers, Environment and Urban Systems - An International Journal. Special Issue on LBS and UbiGIS. Pegamon Press, Elsevier. 2005.
[21] R. Malaka. Deep Map: The multilingual tourist guide. In: Proceedings of the C-STAR workshop. Schwetzingen, 1999.
[22] http://www.di.uniba.it/.
[23] OWL: http://www.w3.org/2004/OWL/.
[24] UbisWorld: http://www.u2m.org/.
[25] M. Horridge, H. Knublauch, A. Rector, R. Stevens, C. Wroe. A Practical Guide To Building OWL Ontologies Using The Prot´eg´e-OWL Plugin and CO-ODE Tools.Edition 1.0. The University Of Manchester. August 27, 2004.
[26] J. Herlocker, J.A. Konstan, A. Brochers, J. Reidl, An algorithmic framekork for performing collaborative filtering, Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, 1999.
[27] T.W. Yan, H. Garcia-Molina, Index Structures for Informational Filtering under the Vector Space Model, in Proceedings of the 10th International Conference on Data Engineering, Huston, TX, USA, February 1994.

Towards Situated Public Displays as Multicast Systems

Hans Jörg Müller and **Antonio Krüger** [1]

Abstract. Within organizations diverse asynchronous, unreliable multicast channels to push information chunks already exist. These usually include posters, newsletters, intranet pages, print media and talks. Situated public displays have the potential to complement these channels, because their properties of being situated and public make them superior to conventional information channels in certain situations. In order to present the information that is most interesting to users and thus improve the efficiency of information distribution, we propose to learn a model of the spatio-temporal behaviour of users, find clusters belonging to certain organizational subgroups within this model, gather profiles for some of the users and use the estimated profiles of clusters to optimize the information presentation on the displays.

1 INTRODUCTION

Imagine you are a student staying around the university for most of the day. You read the newsletter, posters, intranet pages and print flyers. However, you do not have a fixed workplace and feel somehow disconnected from the information flows within the institutes and not being part of it. One day a number of large displays are installed throughout the department showing information chunks about ongoing activities within the institutes. As you enter the building in the morning, you have a quick glimpse of about two seconds on one display to see what is going on today. You notice that there is an interesting talk you have not heard of before, there are two student jobs available and a new research project you are interested in is being announced. You consider these public displays convenient because you do not need any hardware yourself to use them. Furthermore they do not interrupt you during important tasks, since you are only wandering in the hallways anyway and only spend a few seconds to access information. The key advantage of the displays is that they provide information tailored to your interests.

2 RELATED WORK

Various situated public display systems, displays that are installed in public spaces and adapt to their temporal, spatial and social context, already exists. The aim of the GroupCast [4] system is to sense which people are nearby and display information related to their mutual interests to spark informal conversations. The CWall System [3] shows information chunks provided by colleagues and emphasizes the importance of the cost/benefit ratio, but focuses on small "Communities of Practice" instead of whole organizations. The Plasma Poster Network [2] resembles a real poster board where anyone could post items and focuses more on intense interaction with the display than our 'passing by' approach. In contrast to these displays that are located in places where many people pass by, Hermes [1] is an example of office door display systems where visitors can for example leave messages to the owner of the respective office.

[1] University of Münster, Germany, email: joerg.mueller@uni-muenster.de

3 COSTS AND BENEFITS OF SITUATED PUBLIC DISPLAYS

One hypothesis of our work is that situated public displays will only be adopted if the benefits they provide are higher than the costs they incur. On the one hand the costs that occur for the user reading a chunk are determined by what he could have done in the same time otherwise (opportunity costs). These costs depend both on the total time people spend looking at the display as well as on the exact moment. When they are interrupted in some important task the costs will be higher. On the other hand, the benefits for a user reading some chunk are determined by the value of the information transmitted. This depends on whether the information is yet unknown to the reader, and whether it is interesting to the reader at the current time and the current place.

Besides increasing the efficiency of information distribution, more advantages can be identified. First, the system can be used by anyone who does not need to have an own workplace, internet access, email, or mobile phone. Second, discussions can be sparked between people who would otherwise not know of their mutual interests. Furthermore, the displays can provide better situational awareness by showing what is going on *right now* in the organization, and organizational awareness by showing what is going on in general, and thereby make people more feeling part of the organization. The displays also enable opportunistic behaviour, so one could just enter the organization, see what is currently going on and participate in those events.

4 SITUATED PUBLIC DISPLAYS AS MULTICAST SYSTEMS

We want to build an asynchronous, unreliable multicast system for information chunks that are created by people within the organization or sensors within the buildings (sources) and adressed to people within the organization (sinks). The information chunks will describe the organizations status and be either dates (time and place related, like lectures), staffing (like job offers), or pure status information (like project status, publications etc.). Displays are placed where many people pass by, and information should be transferred to people during a time window of two to five seconds while they pass the display. The goal of our work is to improve the efficiency of information distribution by providing information that is tailored to the reader. To achieve this, information about the interests and the spatio-temporal behaviour of people will be gathered (sensing), a model of the population of the organization, their interests and spatio-temporal behaviour will be learned (learning), and the information presentation will be optimized according to this model (acting).

5 SENSING

To adapt the information presentation to the users, both the spatio-temporal behaviour of users and their interests need to be measured. We propose using bluetooth to measure the spatio-temporal behaviour for users who opted in to be logged. Despite the relatively low spatial and temporal resolution of this method (a scan takes about 11 sec.), because most users always carry their mobile phone along, we estimate that a huge amount of data can be gathered using this method.

To measure the interests of users we propose using a combination of different methods. We will provide a facility by which users can download content to their mobile phone or forward them to their email adress. Using bluetooth, we plan to measure the time that people spend in front of the display to estimate whether they are reading content. We are also experimenting with video based face detection to measure whether people are looking at the display and thus probably reading content.

We will provide a web site where people can inspect and delete data gathered about them. They will also be able to edit their profiles, and thus enable the system to provide better individualization of information. There is no need to store personal information like names along with the profiles, as they will only be related to a bluetooth address.

6 LEARNING

As with the sensors, we will need two different user models, one to describe the interests of users and one to describe their spatio-temporal behaviour. Regarding the interests of users, we plan to apply bayesian networks to integrate the information from the different sensors. Regarding the spatio-temporal behaviour of users, we consider to use a time geography based approach to describe the trajectories of different users and find clusters among them. To augment the interests models of users for whom few information is available, we intend to use interests models of users within the same spatio-temporal cluster.

7 ACTING

The goal of our work is to provide the information chunks that are most interesting to users of the system to improve the cost/benefit ratio for the users. To achieve this, for each display and each time point the probability that users from a certain cluster are near to the display will be calculated from the spatio-temporal user model. This measure will be weighted with the probability that users from those clusters show interest for the different information chunks. The chunk with the highest expected value will then be displayed.

8 STATE OF THE IMPLEMENTATION

We currently have installed two prototype displays in our department, one in the main entrance and one in a main hallway (see Figure 1). We are about to install a third display in the second entrance. Currently, on the left part of the display information chunks are displayed in turn without further personalization and are provided display time proportional to the number of letters. Outdated chunks are automatically removed. The first display is running for five months, we have about 10 users regularly submitting information chunks and more than 600 users reading them. Up to now, about 100 information chunks have been posted with a WYSIWYG editor implemented in Java. In addition, on the right part of the display information regarding the building like whether the library is open and information regarding the immediate surroundings of the building, like weather, menu of the cafeteria and bus departure times are displayed. On the bottom, a news ticker shows more general current information.

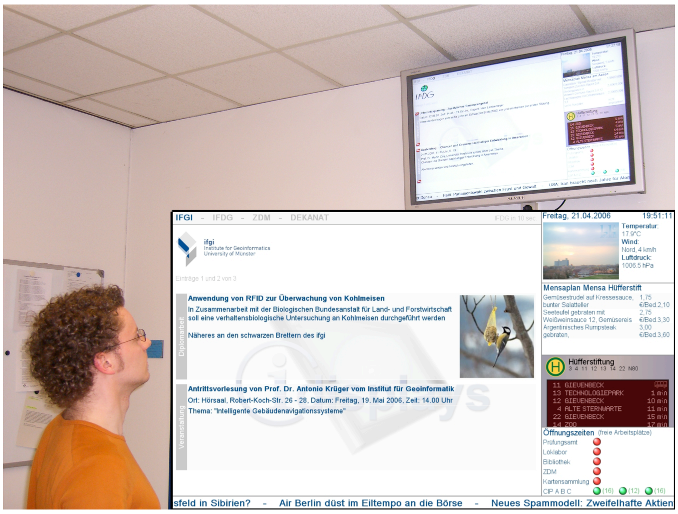

Figure 1. One of the displays installed in the hallway, with information chunks regarding the organization on the left, information regarding the immediate surroundings on the right and a newsticker with general information on the bottom.

9 FUTURE WORK

We presented an approach to improve the efficiency of situated public displays interpreted as an asynchronous, unreliable multicast channel. As next steps we want to finish both the implementation of the sensing hardware and the interaction with the displays with a Java MIDlet on the users mobile phone and gain first experiences with collecting data. We want to further explore the interest model and the spatio-temporal model and implement them. We want to determine how clusters in the spatio-temporal model can be found. Finally, we want to determine how the user model can be exploited to maximize efficiency of information distribution within an organization.

ACKNOWLEDGEMENTS

We would like to thank Oliver Paczkowski and Oliver Meyer for implementing the display prototypes.

REFERENCES

[1] K. Cheverst, D. Fitton, and A. Dix, ‘Exploring the evolution of office door displays’, *Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*, (2003).

[2] Elizabeth Churchill, Les Nelson, Laurent Denoue, Paul Murphy, and Jonathan Helfman, ‘The plasma poster network’, *Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*, (2003).

[3] Antonietta Grasso, Martin Mühlenbrock, Frederic Roulland, and Dave Snowdon, ‘Supporting communities of practice with large screen displays’, *Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*, (2003).

[4] Joseph F. Mccarthy, Tony J. Costa, and Edy S. Liongosari, ‘Unicast, outcast & groupcast: Three steps toward ubiquitous, peripheral displays’, *Lecture Notes in Computer Science*, **2201**, 332+, (January 2001).

Exchanging Personal Information

Christian Wartena, Peter Fennema and **Rogier Brussee** [1]

Abstract. A growing number of organizations stores information on their customers. Both organizations and customers could benefit from exchanging these data more easily and safely. In the present paper we show how Liberty ID–WSF can be used to exchange personal details between organizations in a trustworthy manner that respects privacy and allows the use of multiple identities. The resulting architecture can be used in several environments, including mixes of net based and mobile devices. It provides interaction patterns and interfaces to data consumers, data providers, systems bridging the syntactic and semantic mismatches of user data to increase interoperability and systems that manage the accessibility and privacy of the data.

1 Introduction

A growing number of organizations stores information about their customers. In many cases organizations store much the same data, although there may be a differences in the way the relevant information is represented. Thus companies, governments as well as customers could benefit if personal data could be exchanged more easily. However interchanging personal information immediately raises privacy issues, so such an interchange must be done safely and trustworthy. This is particularly true when information is exchanged in an ad hoc way, and when people use multiple identities i.e. in ubiquitous computing. There are some obvious advantages to direct exchange. People are annoyed by being asked for the same data over and over again. For other personal data, it might cause trouble for people to find them. A further advantage of directly exchanging data is that personal data can be considered as statements about a person. In such cases it matters who makes this statement. Thus a data consumer may require that data comes from a source he considers authoritative.

Consider an example. In Holland, a mortgage is often coupled to a life insurance. Moreover, it usually comes with the requirement to take a fire insurance on the mortgaged property. Thus for the single transaction of a mortgage there are at least three parties involved: the bank, the life insurance and the fire insurance company. Each of these needs information about the mortgage taker. However, not all information can be shared between them equally: the life insurance company is allowed to request some medical data while the bank is not. Conversely, only the bank has access to the data of the BKR (Bureau Krediet Registratie, the institute legally in charge of keeping track of debts of Dutch citizens), a further party that is involved. Each of these parties can ask the same information or they can share some of the information subject to consent of the mortgage taker.

In the following we will often refer to personal data as *profiles*. The person the data is concerned with is called the *profile principal*.

As already becomes clear from the example, there is a number of problems that keeps organizations away from exchanging profiles:

[1] Telematica Instituut, Enschede, The Netherlands, email: {christian.wartena,peter.fennema,rogier.brussee}@telin.nl

- A profile principal might be offended if data given to one organization suddenly show up somewhere else.
- Exchange of data is not allowed by law or explicitly forbidden by the profile principal.
- Profiles used by different organizations and applications are stored and coded in different ways.
- Profile principals have to be identified and a profile principal may use different identities for different organizations.
- The party that has the requested information has to be found.
- There is no suitable infrastructure for exchanging profiles.

In the present paper we concentrate on the last point, taking the other points into account where possible. In the next section we will sketch an infrastructure for the exchange of profiles in which there is place for components that address the other problems. In section 3 we finally compare our approach to other proposals.

2 Architectural support for profile exchange

We will sketch an infrastructure based on the Liberty Identity Web Service Framework (ID–WSF, [16]) to enhance the exchange of profiles that we will call the *Universal Profiling Infrastructure (UPI)*.

We will not presuppose familiarity with the ID–WSF and explain the main concepts we use. We will start with an investigation of the components needed to exchange profiles, concluding that Liberty ID–WSF is indeed a good starting point. However, we will also see that the Liberty ID–SIS Personal Profile Service ([10]), the ID–WSF service for exchanging profiles defined by the Liberty Alliance, has some limitations that we believe make it less versatile than our approach.

2.1 Arguments for a Distributed Approach

2.1.1 Distributed vs. Central User Models

We believe that for an architecture to be successful, it must be based on open standards and put as few requirements to participate on services as possible. We therefore assume that each participating service provider maintains its own user models. It follows that the idea of a a single profile for each person on a central 'user model server' ([11]) shared (and evt. maintained) by all services has to be abandoned. Examples of systems using a centralized view on profiles are DOPPELGÄNGER ([14]) or Microsoft Passport ([5]). Such an architecture might provide a solution for sharing and exchanging data in restricted environments such as e.g. learning environments ([12]). However, in general, we cannot expect all organizations to share sensitive or fiercely protected data in a single repository beyond their immediate control. As a consequence, in our approach all organizations involved maintain responsibility for their own data. In particular it is their own responsibility to assure the correctness and consistency of their profiles. They also need to protect the privacy of the profile's subject themselves and not disclose more than is necessary and consented by the user. This principle can be seen as a variation of the "law of user control and consent" and the "law of minimal disclosure" ([2]). Finally it means that trust becomes an important issue, and that data should be considered as a set of claims for which it is important what the source is.

An interesting alternative approach is sketched in [9]. Here user models are stored in a central repository (but profile principals might have copies of their data and they can share them with profile consumers). However, the data are signed by a trusted (third) party that guarantees the authenticity of the data. Now the only disadvantage for a data providing organization is that she has no longer control over the distribution of the data.

2.1.2 Federated Identity

Having committed ourselves to a distributed view on user models, we are almost forced to assume *federated identity* for profile principals. We say that two identities are federated if they are (explicitly) identified. However, we hasten to say that in an environment with federated identity not all identities that can be federated need to be and that there need not be one single party with a complete overview of all federations. Moreover, federation can be done in an ad hoc process.

An important standard that deals with federated identity is the Security Assertions Mark–up Language (SAML, [3]). Liberty ID–WSF is an architecture to find services that uses federated identity as well and is build on top of SAML. Now it would be natural to consider the exchange of personal information as an interaction between a web service consumer and a web service that provides information about persons. Doing so, we would expect to solve most problems concerning identities and finding information providing parties as well as the main problems with security and the exchange of data. We will first discuss how we believe an architecture for profile exchange should be like. We will then see to what extend this fits the Liberty ID–WSF.

The core SAML and Liberty ID–WSF specifications define protocols for message flow and the messages that have to be exchanged. This in turn is highly determined by the roles that are defined. We will therefore discuss our architecture on the basis of the roles we have identified. We will neither make assumptions on the concrete implementation or deployment of roles nor assume anything on the content or structure of the profiles, since we believe this is the responsibility of the cooperating parties.

2.2 Roles

Besides the obvious roles of profile provider (profile disclosure service) and profile consumer we distinguish a number of further roles in the process of exchanging profiles.

2.2.1 Identity Provider

As discussed above we assume that user data are not stored at a central place but rather distributed among the parties that collected and own the data. In this distributed scenario it would be strange to assume a common identity of profile principals for all services. Rather we should build upon federated identity as argued above. Thus, a very important role in our architecture is that of *identity provider*. As said before, there need not be one central identity provider. In deployment contexts where all parties involved use the same representation (e.g. an email address), identity mapping is not required. In general, however, each party has a local representation of an identity. They can only communicate about a person (i.e. exchange profiles) if they can somehow relate these local characterizations of identity. For this purpose they can call an identity mapping function.

In addition, the identity provider is responsible for the single–sign–on aspects of the protocol. The profile consumer signs in at the identity provider. There is no need to sign in to all other services used in the course of the protocol, since the services all belong to the circle of trust of the identity provider. In order to obtain this functionality, the identity provider will give the profile consumer a certificate that allows him to use the other services.

2.2.2 Profile Discovery Service

The next role we distinguish, is the *profile discovery service*. The profile discovery service can be called by a profile consumer to find out which party might have the information he is looking for, i.e. information on a certain identity, eventually even restricted to the requested type of data. This role is almost identical to the Liberty ID–WSF Discovery Service ([15]). The discovery service is able to fulfill requests for a certain type of service (in our case giving personal data) that is available w.r.t. a certain identity. This call has an optional parameter that can be used to tell the discovery service to restrict the list of answers to service instances that provide data of a certain type, e.g. financial data. The Liberty ID–SIS Personal Profile Service uses the discovery service in exactly this way. The service may be called if necessary, but needs not to if the profile consumer already knows which party can provide the requested data. This is in fact a common situation. In the example in the introduction, if someone wants to get a mortgage the bank wants to know whether the mortgage taker has other active debts. In this case (in the Netherlands) the bank wants to get this information only from the BKR.

The function of the profile discovery service is very similar to that of the identity provider. The identity provider has basically to maintain a table to look up how an identity from a certain party is represented at an another party. Similarly, the profile discovery service has to maintain a table to look up which parties can provide what sort of identity information. Thus in practice it would be a viable and good idea to implement one service that plays both roles. In fact, this is also recommended practice for the Liberty ID–WSF Discovery Service and the identity provider in the ID–WSF context in general ([15, p. 14],[19, p. 10]).

2.2.3 Translation Service

An important role without parallel in the projects of the Liberty Alliance is the *profile translation service*. In our distributed architecture for user modeling this is an essential component. Since we did not want to assume that all parties use a common identity representation or one common profile for each principal we will have to assume that parties can use different languages and different conceptual schemas to code a profile. Even if two parties agree on the syntax to code a profile it is still likely that they cannot interpret each others data. Figure 2.2.3 illustrates how an easy piece of data for the address of a person can be expressed in different ways. To get an impression of the different standards in use to describe basic facts about persons we refer to a report of the Fidis Project ([7]).

The translation service will convert data from one representation into the other as far as possible. This might be done in several ways.

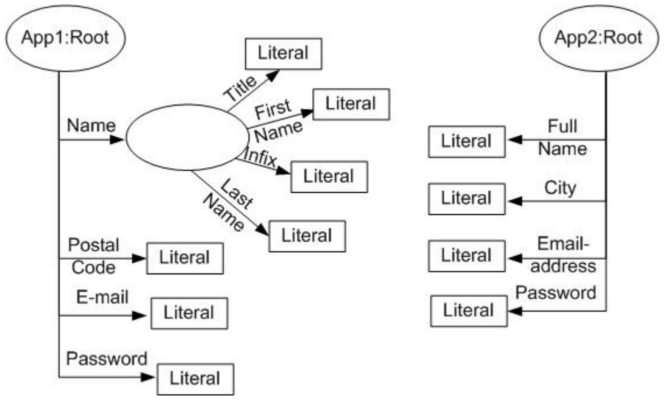

Figure 1. Two examples of user model schemas

The first possibility is that the service has knowledge of both profile schemas and how they relate. For example, if both profiles are expressed using concepts from formal ontologies and a mapping between the ontologies is provided. Note that an translation service can be designed such that parties can provide it with new mappings. The second possibility is that the service uses some heuristics to find out how two schemas have to match. The first approach is more reliable and will be favored if data exchange between two parties is done on a regular base and correctness of the data is important. The second approach is valuable e.g. in the context of ad hoc exchange of data as it is frequently the case in ubiquitous computing. Again, two parties might also agree on the language they speak (and in fact in many cases that would be the best choice), but in general we cannot assume this. Note that an adequate translation server (i.e., one that offers the required translation direction) can be found using the regular ID–WSF discovery service. The development or training of a translation service is out of the scope of this paper, but a central topic in the AlterEgo project, in which the UPI–Architecture was developed. An example of a profile translation service is described in [18].

2.2.4 Applicability of ID–WSF

We conclude that the Liberty ID–WSF fits our needs quite well and the components of the ID–WSF are all required. Within this framework we can define the profile disclosure service and the translation service. An overview of the components and their relation to Liberty ID–WSF and SAML is given in Figure 2.2.4. In addition we have to specify the messages and protocols that are required to interact with the translation service.

2.3 Profiles and messages

As said above, the majority of the messages needed for data exchange between the services described above, are specified by Liberty ID–WSF. Two important messages that have to be defined in the context of the UPI are the profile query and the profile response message. The message specification contains obvious elements like information on the issuer of a statement and the date and time of an assertions, but it says nothing on the structure of the profile request and the profile itself. The messages also offer a place to make meta–statements about the profile. For example, it might be useful to say whether information might be incorrect due to an heuristics–based translation.

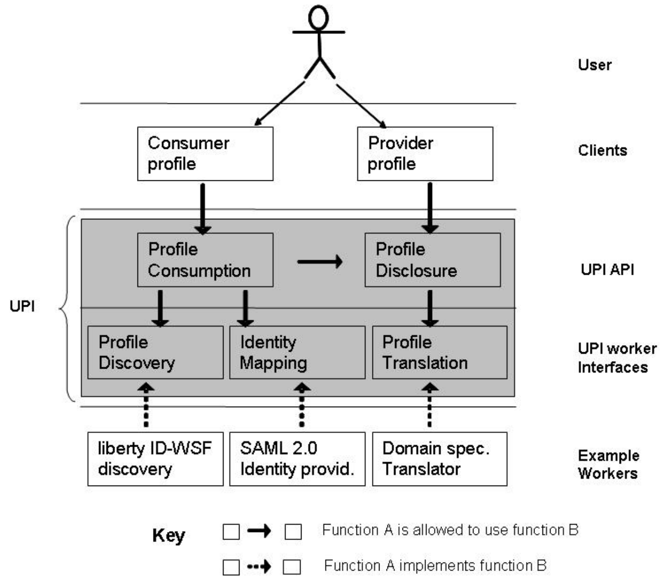

Figure 2. UPI Functional Architecture

Finally, note that a profile request contains a profile query. Thus, a profile provider only needs to expose the part of the profile that is requested. It is not necessary to reveal a complete profile every time some information is needed.

2.4 Collaboration

In the following we describe the steps required to obtain profile information on a profile principal. We assume the most difficult scenario, i.e. there is no common characterization of identity, the potential information provider is unknown and the parties are using different profile schemas. However, we will assume that there is one service that acts both as identity provider and discovery service.

2.4.1 Example 1

Consider the following scenario. In the Netherlands, employers are supposed to pay their employees $1/6$ of the cost of their children's day care. Suppose John is an employee of the MegaBuck company and has a contract for day care at the MiniMouse day care center. The MegaBuck salary administration wants to know how much John pays MiniMouse a month.

First MegaBuck has to sign on at an identity provider. We assume that MegaBuck has an account at all other services and that these services belong to one network of trust. Thus, MegaBuck can use all other services without (explicitly) signing on, since MegaBuck's credentials are propagated through the single-sign-on mechanism as defined by SAML.

Now MegaBuck can send a request to the profile discovery service to get the address of the service providing details on John's contract for day nursery. The response includes a (possibly empty) list of services along with a representation of John's identity at that service and with credentials to use that service. Note, that in case the identity provider and discovery service are separated a lot more communication would be required. MegaBuck now decides which of those profile disclosure services he considers as authoritative, in this case MiniMouse, and sends a message with a request for the data he wants to get and the credentials he got from the discovery service/identity provider. Since the request is formulated in a vocabulary the disclosure service is not able to work with, the message has to be redirected to the translation service. If necessary the answer (the information on John's nursery contract) is sent to the translation service as well in order to translate it into the original vocabulary. Finally, the (encoded) data are sent to MegaBuck.

2.4.2 Example 2

In the examples above, both in the case of the day care center and in in the case of the mortgage, we have used federated identities and a trusted third party, which is aware of this federation, playing the role of identity provider. In some cases this is a worthwhile scenario, especially if identities were already federated to enable single–sign–on to begin with. In other cases however, federating identities might not be advisable. In such cases the profile principal could play the role of identity provider himself. Thus he will in fact be able to link two of his identities once–only.

Consider the following scenario. John is a scientist at the University of Harderwijk. John is visiting a conference at Riva del Garda. The conference offers a service for selecting the talks that might be interesting to someone given his areas of expertise. After registration John gets an account for this service. John's university has a list of John's expertise areas. However, John's university and the conference organizers don't belong to a common circle of trust and John's identities at his university and at the conference are not federated. Now John can act as an identity provider. On receiving the request for expertise areas from the conference, (a program on the computer of) John will give the conference service the address of the service of his university along with a certificate with an encrypted representation of his identity that gives the conference service sufficient rights to temporarily use the service of John's university.

2.4.3 Exchange Protocols

Beside the protocol for obtaining profile information from another party illustrated above, there are some further important protocols, notably the protocol needed to federate an identity and the protocol for registering a disclosure service at a discovery service. The latter can be done according to the ID–WSF specifications, the former is realized by the standard ID–FF or SAML mechanisms[2]. This can be done without problems since ID–WSF is defined on top of these specifications.

2.5 Protecting Privacy

Up to now, we have ignored the important privacy aspect of the exchange of personal data. The certificates sent to the disclosure service do not guarantee privacy. They only allow parties to use the disclosure service but do not make assertions on which parties are allowed to see what data. For acceptance of exchanging profile information informed consent is very important ([17]). In our vision this does not necessarily mean that the profile principal has to explicitly approve each transaction. Eldin e.a. ([8]) describe a system that is able to decide whether data can be delivered on the basis of a simple policy. One of the input parameters that their system (and probably almost each comparable system) needs, is the purpose of the request of the profile consumer. For example, a profile policy might state that some piece of data can be given to a commercial party provided it is not used for marketing purposes. This means that the request message from the profile consumer to the disclosure service should contain a (possibly empty) list of purposes.

In some cases, a policy could state that an explicit permission of the profile principal is required. This complicates the situation since the principal is a party that up to now was not involved in the process. Without a mechanism for the principal to give his consent, the data should not be shared. However, in many cases the profile consumer is a service that is acting on behalf of the principal. For example, one could imagine a scenario in which the principal connects to a service like a personalized recommender system. Such a system might send a request for personal data to some profile disclosure service that the principal might generally prefer not to make public. Since this is a general pattern showing up with identity based services, the ID–WSF describes how the disclosure (or any other) service and the web service consumer can cooperate to redirect the principal to the disclosure service, allowing the disclosure service to interact with the principal ([1]). The principal can then give permission to the disclosure service to give the profile consumer access to his resources.

2.6 API

The messages and protocols described above provide a solid and secure framework for the exchange of information about persons. However, handling of the messages is a rather complicated task that requires a detailed knowledge of all specifications to implement the services. In order to make it easier to develop profile disclosure and consumer services we started to develop a reference implementation including an application programming interface (API).

3 Conclusion

3.1 Comparison to Other Work

3.1.1 Liberty Personal Profile Service

We already referred to Liberty Personal Profile Service (ID–SIS-PP) several times. As has become clear, it is very similar to our approach. However, it is strongly focused on a small set of common data such as addresses, affiliation, etc. There is almost no support for data beyond this set. There is nothing like a translation service (though it might be compatible with an ID–SIS–PP service) and there is no way to specify other than the standard data in a discovery request.

Finally, note that the Liberty Alliance puts much effort in the development of open standards but is not pursuing open source implementations. However, there do exists some open source implementations of SAML.

3.1.2 Microsoft Passport Network

The .net Passport system, later renamed Passport network is a system for federated identity and single sign on widely deployed by Microsoft ([5]). It is well integrated with the authorization mechanisms of Windows XP, and is used by all users of their network services such as Hotmail and Messenger. The original .net passport system had a centralized server to store various user and user behavior related properties. Moreover it was designed to serve as a universal authentication framework for hailstorm, the .net e-commerce and enterprise integration web services framework. However, .net passport

[2] For a discussion of the relationship between ID–FF and SAML see e.g. [13].

had poor uptake outside of MSN services because neither service providers nor users felt comfortable with a centrally controlled system. As a result Microsoft restricted the use of Passport Network to use in the MSN services

3.1.3 Microsoft Infocard

Microsoft has positioned the Infocard architecture ([6]) as a "meta integration" framework to integrate federated identity systems using active server, Kerberos, SAML, LDAP etc, apparently based on the web service stack (including SAML, [4]). Conceptually, the Infocard architecture is based on "the laws of identity" ([2]). The Infocard architecture distinguishes three roles: identity providers, relying parties (profile requesters) and subjects (Profile principals). Identity is treated as a set of claims on a subject. Identity providers make such claims and provide proof on them which relying parties may or may not accept. Important components are

1. a negotiation component where parties can negotiate on the claims that they want to know or accept, somewhat similar to the discovery service in the UPI.
2. a claim transformer that translates claims trusted and understood by one party in terms of the other, similar to UPI translation service.
3. an identity management component that allows users to manage identities which will be deeply integrated in the Windows operating system and GUI to avoid phishing attacks.
4. a protocol encapsulator to bridge different identity federation protocols.

3.1.4 Higgins

Higgins is an open source answer to Infocard backed by a.o. IBM and Novell. The project is based on the eclipse framework but seems still to be in the first stages. Some information about the project can be found at http://spwiki.editme.com/Higgins. It has a strong focus on user control of identity. Theoretically speaking this position is somewhat different form ours, since we emphasized the responsibility of the organizations collecting, maintaining and using the data. However, these organizations will not obtain data about their customers if there are no possibilities for the profile principals to specify policies that are respected. Thus the policies are an essential component in our approach. On the other hand side, organizations will not support a framework for exchanging data if this would imply loss of control over the data. Thus Higgins and the UPI might turn out to be very similar in this respect.

The Higgins project intends to develop a number of plug-ins that can be used by services that want to share profiles. All issues concerning identities, authorization, message handling, etc. are encapsulated by the Higgins plug-ins and hidden to the core application. These plug-ins thus are very similar to the UPI API. It is not clear whether Higgins builds upon open standards, like SAML and ID–WSF, to solve these problems, or whether it uses a custom solution.

3.2 Final remarks

In comparison to other proposals we impose very few conditions on the parties that want to participate in a network of profile exchanging services. We did not assume a common identity for principals and we did not assume a common language for the representation of profiles. This might seem unnecessarily complicated in case only standard data like names, addresses and phone numbers are exchanged. As soon as we move to more specialized domains however, it will become more and more difficult to agree on a single representation. Of course it is not possible to define fields for all kinds of information in advance. Other solutions, like the ID–SIS–PP recognized this as well and offer a field for 'other' information, but without any support for handling it. In contrast, in our proposed solution, the 'other' information has become central and we offer a way to handle this information by including a translation service into our framework.

The price we have to pay for this distributed approach is that we need translation services and identity providers, and that all the parties have to register at an identity provider. However, since we were able to do this using standard mechanisms, we believe that the advantages of our low requirements are greater than the disadvantages of the additional protocols.

The fact that we did not commit to one common representation of profiles has one drawback for which we do not currently have a final solution. In our discussion of the discovery service we suggested that this service does not only find profile disclosure services that are relevant w.r.t. to a certain principal but also w.r.t. a certain query. Thus, a profile disclosure service should supply the discovery service with some topics it has information about. When a consumer consults the discovery service it could also specify the topics he is interested in. This means that we either have to commit to one single set of topics or we have to build in the possibility that the discovery service calls the translation service to translate the topics.

A further issue concerning the translation service that needs to be worked out further is the privacy of the translated data. The translation request might contain an identifiable set of data. If this is the case, the translation can only be done by a party that is trusted by all other parties involved.

4 Acknowledgements

This work was carried out within the Alter Ego Project. Alter Ego is part of the MultimediaN program (http://www.multimedian.nl) which is sponsored by the Dutch government under contract BSIK 03031. Alter Ego participants are IBM, Technical University Delft, Technical University Eindhoven and the Telematica Instituut. We would like to thank the members of the Alter Ego team for their contribution to the UPI and helpful comments on this paper. We would like to thank an anonymous reviewer for drawing our attention to [9].

REFERENCES

[1] Robert Aarts, editor. *Liberty ID-WSF Interaction Service Specification*. Liberty Alliance, 2004. http://www.projectliberty.org/specs/liberty-idwsf-interaction-svc-v1.1.pdf.

[2] Kim Cameron. The Laws of Identity, May 2005. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/lawsofidentity.asp.

[3] Scott Cantor, John Kemp, Rob Philpott, and Eve Maler, editors. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. Oasis, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[4] Microsoft Corporation. A Technical Reference for InfoCard v1.0 in Windows, 2005. http://download.microsoft.com/download/5/4/0/54091e0b-464c-4961-a934-d47f91b66228/infocard-techref-beta2-published.pdf.

[5] Microsoft Corporation. Microsoft passport network privacy statement, 2005. https://accountservices.passport.net/PPPrivacyStatement.srf?ru=https://accountservices.passport.net/ppnetworkhome.srf.

[6] Microsoft Corporation. Microsoft's Vision for an Identity Metasystem, 2005. http://msdn.microsoft.com/winfx/reference/infocard/default.aspx?pull=/library/en-us/dnwebsrv/html/identitymetasystem.asp.

[7] Thierry Nabeth e.a. Models. Technical report, FIDIS, october 2005. http://www.fidis.net/fileadmin/fidis/deliverables/fidis-wp2-del2.3.models.pdf.

[8] Amr Ali Eldin, René Wagenaar, and Harry Bouwman. Towards dynamic control of users privacy. In *5th IBIMA Conference on The Internet & Information Technology in Modern Organizations*, december 2005.

[9] Michael Hitchens, Judy Kay, Bob Kummerfeld, and Ajay Brar. Secure identity management for pseudo-anonymous service access. In Dieter Hutter and Markus Ullmann, editors, *Security in Pervasive Computing*, volume 3450 of *Lecture Notes in Computer Science*, pages 48–55. Springer, 2005.

[10] Sampo Kellomäki and Rob Lockhart, editors. *Liberty ID-SIS Personal Profile Service Specification*. Liberty Alliance, 2005. http://www.projectliberty.org/specs/liberty-idsis-pp-v1.1.pdf.

[11] Alfred Kobsa and Wolfgang Pohl. The user modeling shell system bgp-ms. *User Model. User-Adapt. Interact.*, 4(2):59–106, 1995.

[12] I. Machado, A. Martins, and A. Paiva. One for all and all in one. In Judy Kay, editor, *User Modeling*. Springer, 1999.

[13] Paul Madsen. SAML 2: The Building Blocks of Federated Identity, 2005. http://www.xml.com/pub/a/2005/01/12/saml2.html.

[14] Jon Orwant. Heterogeneous learning in the doppelgänger user modeling system. *User Model. User-Adapt. Interact.*, 4(2):107–130, 1995.

[15] Jonathan Sergent, editor. *Liberty ID-WSF Discovery Service Specification*. Liberty Alliance, 2004. http://www.projectliberty.org/specs/liberty-idwsf-disco-svc-v1.2.pdf.

[16] Jonathan Tourzan and Yuzo Koga, editors. *Liberty ID-WSF Web Service Framework Overview*. Liberty Alliance, 2004. http://www.projectliberty.org/specs/draft-liberty-idwsf-overview-1.0-errata-v1.0.pdf.

[17] Thea van der Geest, Willem Pieterson, and Peter de Vries. Informed consent to address trust, control, and privacy concerns in user profiling. In Alfred Kobsa and Lorrie Cranor, editors, *Privacy Enhanced Personalisation, PEP 2005, Edinburgh, Scotland, July 25, 2005, Proceedings*, pages 23–34, 2005.

[18] Kees van der Sluijs and Geert-Jan Houben. A generic component for exchanging user models between web–based systems. *International Journal of Continuing Engineering Education and Life-Long Learning*, 16(1/2):64–76, 2005.

[19] Christine Varney. Privacy and Security Best Practices. Technical report, Liberty Alliance Project, november 2003. http://www.projectliberty.org/specs/final_privacy_security_best_practices.pdf.

User Modeling in a Distributed Multi-Modal Application

Andreas Zimmermann[1], Andreas Lorenz[1]

Abstract. In distributed multi-modal applications several steps of adaptive processing are spread across devices, including client/server-architectures implementing the application logic as well as different interaction devices on both the input (e.g. touch screen, microphone) and the output side (e.g. display, headset). This paper introduces the MICA-project and describes the interplay and communication between its distributed components.

1 INTRODUCTION

MICA is a project on behalf of and in co-operation with SAP. The project started in December 2004 running until November 2006. The goal of MICA is to support humans in their working environment in a natural and unobtrusive way and to proactively help them completing their daily tasks.

In a first phase MICA is implemented in a warehouse scenario to support warehouse workers in the picking process. The warehouse scenario poses a real challenge for multi-modal interaction. In a warehouse the workers are working with their hands thus it requires a hands-free support. The environment is often very noisy and the light conditions might change as well so the interaction might use different modalities according to the needs of the current situation and task. Warehouse men often have to work under time pressure requiring a very responsive system.

The goal regarding interaction in MICA is to provide a combination of explicit and implicit interaction methods in blue collar environments. Unlike the IBM MAGIC system, which uses gaze tracking for the prediction of cursor movement [8], we favour the combination of speech and pen input with user movement in a physical environment.

In contrast to the LISTEN [10] project we face situations in which the spatial relations of objects to each other are changing dynamically and need to be monitored and interpreted in real time. In MICA we combined a low precision WLAN tracking with fine grained UWB Tracking, RFID and camera based motion tracking to be able to determine the spatial and situational parameters. On the one hand the system will be able to identify situations in which help might be needed and react on implicit clues in the interaction like stumbling or search behavior. On the other hand the worker will be able to interact explicitly with the system by approaching objects in the shelf. In particular the combination of implicit and explicit interaction on various modalities will lead to natural blended interaction [3]. It is essential though to improve our understanding of the interactive capabilities that are most important for an automated system to conduct a natural multimodal dialogue.

In this paper, we will first introduce the design of the architecture and describe the role of different clusters in the adaptive processing. In the second chapter we will then illustrate the implications to the distributed user model in the project. At the end we will explain the implementation and specifically focus on the communication between components fulfilling different roles in the adaptive process.

[1] Fraunhofer Institute for Applied Information Technology, Schloss Birlinghoven, 53754 Sankt Augustin, Germany, {Andreas. Zimmermann, Andreas. Lorenz}@fit.fraunhofer.de

1.1 Aspects of System Design

Recent architectures for multimodal applications (e.g. [1], [2]) have especially achieved success in support for sensing the multimodal user interaction. This research results in high-end abilities for recognition and synthesis in common modalities such as speech and handwriting. Additionally, current architectures focus on synchronizing events coming from different devices, such as keyboard, mouse, microphone, etc., allowing flexible handling multimodal interactions. In our work, we will go beyond observing input from the different modalities to also integrate the recognition and interpretation of meaningful user-related and environmental parameters. For example, the interaction of a mobile worker with her device will particularly be determined by her current tasks, goals and situation. From our point of view, the crucial tasks of building and maintaining a meaningful user-model as well as an environmental model are currently underrepresented.

In its framework-definition [9], the W3C introduces the Interaction-Manager for controlling the sequence of exchange for information between the human user and the application functions. Besides the fact that the knowledge in the user- and environment model are the main influencing factors for the selection of the best modality and for adapting the output stream to the particular device used, we think that managing the interaction on the output side should start with the selection of the appropriate content. In the content-filtering process, the availability of output modalities needs to be carefully considered, as well as filtering regarding content. If the worker is currently not wearing headphones, the selection of audio content is not useful, even if it would best fit the worker's information demand. Therefore, we aim at a dialogue management process integrating both the interaction management and the content management part.

1.2 The MICA-Architecture

It is essential to improve our understanding of the interactive capabilities that are most important for an automated system to conduct a natural multimodal dialogue. We need to integrate a wide range of behavioral data from human users interacting with multimodal applications. Such analysis provides concrete hypotheses for directions to improve components of open multimodal context-adaptive systems, as well as the overall design and architecture of such systems, which are briefly described below. In this approach, the contextualization process is divided into four steps [11], each fulfilling a specific role corresponding to a single layer in the architecture: Input, Modeling, Dialogue Management and Output (cf. Figure 1).

The Input Layer recognizes parameters relevant for the operation of the application. Sensors monitor the interaction of the user with the system and measure contextual parameters of the user (e.g. position, heart rate) and the environment (e.g. room temperature, light conditions). The user's interaction with the multimodal interface comprises the interaction modality (e.g. expressed by microphone) and the input data (e.g. the spoken word), and needs to be explicitly monitored. A sensoring subtask combines information from several uni-modal sensors by multimodal fusion. The last step, multimodal recognition, extracts commands from speech or gestures, and implicit feedback patterns, e.g. movement patterns [3]. Beside other sensor-related data, the sensor database contains grammars, vocabularies, templates and patterns necessary for a successful recognition process.

The Modeling Layer semantically enriches the data stream from the sensors by interpretation. Therefore, intelligent modules aggregate information by interpreting the data with meta-information and refining the knowledge-base. The emerging knowledge is stored in a user model (e.g. tasks, knowledge, and cognitive, attentive states) and an environmental model. These two models enable resolving ambiguities by considering the contextual parameters, since the different modes of a multimodal system are not simple analogues of one another and do not involve redundant but complementary information and can be used for mutual disambiguation [7]. The user model particularly determines the content selection and the information about the user's environment influences the adaptation of the presentation to the right form.

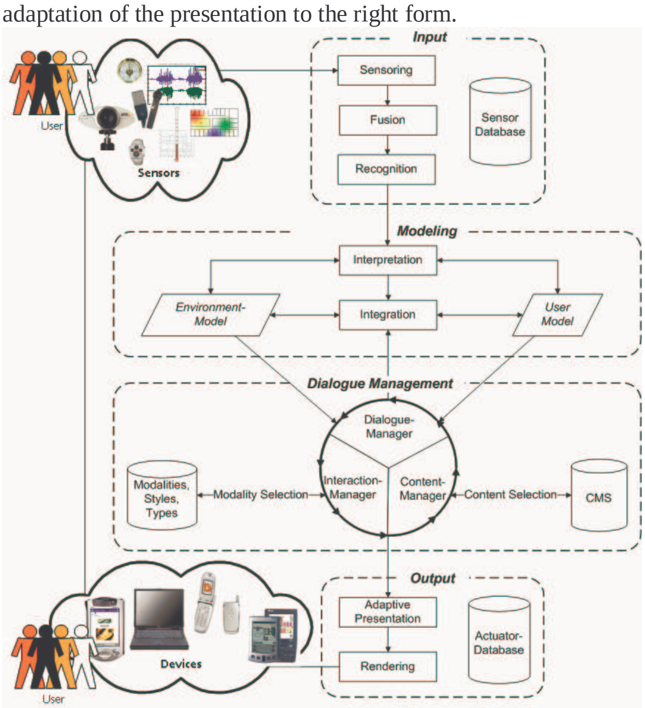

Figure 1 The MICA-Architecture

The Dialogue Management layer comprises dialogue, interaction and content manager that co-operate jointly, in order to take appropriate actions, that base on derived knowledge. The dialogue manager plans, assembles and refines sequences of commands to control the behavior of the system and handle the dialogue with the user. The dialogue manager queries the content manager and the interaction manager and combines their suggestions. The content manager is responsible for the context-sensitive information selection from the content-repository, since it is not enough to just supply content without the consideration of the recipient, her current task, and situation ([10], [11]). The interaction manager creates and traces certain interaction strategies. Additionally, it adapts to certain conditions in the environment or in the individual user context. The adaptive or strategic methods are implemented domain-independently, although the expressivity of a modality might be domain-dependent.

The Output Layer generates appropriate output and handles the connection back to the users. It translates the decisions taken by the dialogue manager into real world actions on a wide range of multimodal terminals. An accurate respond bases on the distribution of output among modalities and media in accordance with the user and dialogue-system contexts, e.g. to coordinate multimodal output between different kinds of devices like PDAs or cell-phones. The presentation of the content is adapted to the different technical capabilities of the targeted devices.

2 User Modeling in MICA

In this chapter, we will first discuss the options of centralized vs. decentralized processing and then we will illustrate our user modeling approach for the MICA-application.

2.1 Centralized Versus Decentralized Processing

For applications like MICA considering multimodal interaction for independent mobile users, monolithic all-in-one approaches seem not to be feasible. For example, sensors of different kinds are distributed independently within the environment; fixed sensors like local tracking-systems or RFID-readers need to be integrated with other sensors that have to be as mobile as the user is (e.g. noise and light sensors). The same holds true for actuators, where static public displays and loud-speakers co-exist with private mobile displays and headphones. Other decisions whether to host all user-models on a central server versus several user model instances on different devices, or whether to centrally control each user-dialogue versus a local dialogue management on the end-device, or whether to centrally provide contents versus a local pre-storing of contents on the device, depend on the right balance between the three aspects:

1. Processing power, memory, and storage of the end-device.
2. Network traffic and runtime delays issued by inter-component communication
3. Synchronization of shared contents and user-related data.

Balancing the task performance between a central powerful server and local small end-devices within MICA, i.e. the discussion of the advantages and disadvantages of centralized and distributed processing, revealed that a mix of centralized and distributed processes would be most appropriate. The advantages of centralized processes minimizing network traffic, which is already considerably high due to a amount of sensor data from tracking devices, RFID-Readers, Camera etc, would be best combined with a reasonable amount of processing on the devices, providing more flexibility and better scalability. In connection with the discussion of centralized versus distributed processing we developed a brokering model for sensor and actuator data flow introduced in [5].

2.2 User Modeling

The MICA system uses seven sensors and the interaction devices for direct input from the user. The sensors detect articles by using RFID readers built in the shelves and on the trolley, locate the user by using the WLAN-based tracking systems of Ubisense and Ekahau, analyze the user's search behavior by a small camera and measure attributes of the user's environment through the headset microphone and a light sensor. In addition, the user provides information about the current task, devices in use and the order through explicit selection.

For the warehouse it is important to keep all data consistent. First, all articles in the warehouse and the (open) orders consisting of articles are written to a database on the server. Additionally, workers have a static profile with worker-id, name, role in the process and qualification inside the database. At runtime, distributed components fill in dynamic worker-attributes.

Using her id-card, the worker logs into the system and the client requests the worker-profile from the server. If the id is valid, the client registers the worker with her devices: The current trolley with tablet-PC, the local camera, the headset with microphone and earphone, and both tracking-tags are connected with the worker-id. From now on the client is able to identify the behavior of exact the current worker. The identification is of particular importance to filter all information coming in from other entities that relate to the current worker or her devices.

In the next step, the client requests all open tasks the worker is able to perform from the server. From this list, the worker selects any number of tasks (selection is done by hand, usually depending on the box size, because all selected tasks have to share the trolley). Each task consists of a list of article-amount pairs and each article is associated with a position, a description and an image. The task profiles are retrieved from the server and the client responses the selected tasks to the server in order to keep the task-manager up to date. If another worker is logging on, the tasks that are already in execution are not transmitted to any other worker again.

When the worker is moving around, her absolute position is tracked by two independent tracking systems. Both systems report changes to the server where a sensor fusion component combines the positioning data in order to get the most appropriate value for the user position. The Ekahau position information is available in the entire lab but the Ubisense system only covers one room of the lab due to costs reasons. Since the Ubisense system is more precise than the Ekahau system [6], the Ubisense tracking data prevails, if the user is in the range of this system. The server posts the fused tracking data to the client(s) where patterns of the worker's movement are recognized by the client associated with the tracked worker. Combined with the analysis of head-movements (done by image-processing from the camera-input on the client), these movement patterns are an import input to the MICA help system because it allows for pro-active help, if the user might be looking for something [4]. Since determining searching behavior of a specific worker is only useful for her specific client, we decided not to post the movement pattern and the head-movements to other entities. By the local image-processing and decision finding for presenting help pro-actively, we experienced a drastically slow-down of the tablet-PC hosting the client application.

Beside the determination of searching behavior, the navigation support on the client shows the position of the worker in relation to the positions of the articles to be picked up as well as the route the worker should walk. Because the route is highly dependent on the articles left, the route-generation is done on the server for all workers. In future improvements this might also support collaboration of workers where the server can generate routes where workers will meet. For now, the server-side route generation at least relieves the client from the resource-intensive task.

When the worker takes any article from the shelf to the trolley, the RFID-readers post corresponding events and the client who is interested is performing the message, i.e. requesting the object (box, article) associated with the id and updating the display. If the object was a valid object placed on the right place on the trolley, the client will notify the server in order to update the database. Which client is processing the RFID-event depends on the id of the message-sender: From the login-procedure the trolley and the readers on the trolley were connected with the user.

Coming back to the client-side, local noise and brightness are locally acquired and stored on the mobile devices. Both attributes are needed to adjust the output at the local device and do not have any meaning for other entities. Therefore, they are not distributed on the network but only used to sign over commands from other entities, e.g. to display graphical output in bright environment or to display audio-output in noisy environments.

3 Implementation

In this chapter we will illustrate the implementation issues of the MICA system with a detailed view on the communication between the components.

3.1 Server

The software architecture of the MICA server implemented the modular approach of the system architecture (Figure 1). The server receives messages from the sensors and from the clients via the input bus. Messages are evaluated and forwarded to the component responsible for handling that specific kind of message. This message forwarding is also realized with instant messages (see below) to make each component as independent as possible.

A set of rules have been specified and put into the integrated Context Management System (CXMS; [11]). The CXMS then generates messages to trigger other parts of the system e.g. the Dialogue Manager, and the Interaction Manager. This is the core of the pro-active behavior of the system, whereas the Dialogue Management composes messages that tell the receiver what information to display and in which modality. This message will thus still contain all available modalities and it is up to the client to decide which modality will be used for output. This is necessary because the user could be in a situation that is unknown to the server and prevents outputting in a certain modality, e.g. if the environment of the user is very noisy and therefore she cannot understand an audio message.

3.2 Client

The client software serves as the central instance for the interaction of the user with the system and was designed to be as flexible as possible. It should be easily extensible, provide flexible mechanisms for displaying information in different modalities and adapt to the context of the user.

Each of the components authenticates itself against the Jabber-server (which is described in the next chapter) and subscribes to chat rooms on its first initialization. The client will listen to incoming messages and respond to them. For information rendering the client will receive a message consisting of a set of possible modalities and which content is suitable and then decides on how to render it. This decision is taken by the component itself because it has to take the results from its local sensors into account. It might be the case that its local sensors report a high noise level but the server requested to play an audio file that is actually no longer audible for the user. Therefore it is useful that the client component decides to display the content on the screen or inform the user that there is a content that cannot be played right now.

3.3 Communication

As the basis for the communication between distributed components, we decided to rely on a standards-based instant messaging server based on the XMPP protocol also known as Jabber[2]. Jabber is a set of streaming XML protocols and

[2] http://www.jabber.org

technologies that enable any two entities on the Internet to exchange messages, presence, and other structured information in close to real time. The Jabber-server acts as a message distribution center which handles authentication of users, multi-user-chat room management, message distribution and caching. Jabber-clients can subscribe to multi-user-chat rooms where everyone receives a message sent to this room.

These multi-user-chat rooms are used for distributing messages that are potentially valid for more than one receiver e.g. location-sensor-data. The room-name defines what kind of messages are distributed in there and each component interested in that kind of information has to subscribe to that room. Using distinct chat rooms ensures that the information sent to a group chat is sent to all registered listeners. Beside the opportunity to use Jabber's group chat-API, using different chat rooms additionally supports structured information flow, because each specific room hosts data of a specific data type. This allows for efficient messages distribution between all components of the system and is flexible with only minor complexity even in large systems.

4.3.1 MICA Message-Format

To support communication abilities and minimize misinterpretations between different components, a well defined message format is indispensable. The handicap of our first candidate, XML, is that parsing a deeply nested XML-structure with a high number of sublevels might overstrain the capabilities of small devices. The recursive parsing of such a structure could easily force the receiver into a stack overflow. Therefore we decided to cut down structuring without loosing too much power of expression. We finally found a definition fulfilling all above-mentioned requirements in the Attribute-Relation-File-Format (ARFF[3]). Based on this format we defined messages of five types:

- `Command` – A command from a sender to the receiver to do something.
- `Event` – An asynchronous delivery of new data
- `Request` – The sender requests a specific information from the receiver
- `Response` – The answer to a received Request
- `Exception` – Informing other that something went wrong.

Each message is composed of a set of pre-defined fields (building the message header) and an optional set of run-time parameters. To give an idea of such a message, here is an example of the RFID-Event sent by the reader installed on the trolley:

```
@relation event
@attribute from
@attribute to
@attribute reference
@attribute #tagID
@data tabletpc1rfid2,localroomtabletpc1,RFIDadded,
E004010000D5C366
```

Read the message as: "The RFID-reader2 at tabletpc1 informs all listeners registered to localroomtabletpc1 that the tag with the ID E004010000D5C366 is now in the range of the reader". Since it is an event-message, the sender just informs the room-listeners – it won't care about the effects and won't expect any confirmation whether the message was received successfully.

The receiving component independently decides what to do with the information. If the component is waiting for an article to be added it would then query the server if the added tag was an article and nothing else. The request would be sent to the room "tagidentification" in which all client components can post their queries. This room architecture reduces the complexity of the components because the definition which component has to listen in what room is simplified.

[3] http://www.cs.waikato.ac.nz/~ml/ weka/arff.html

4 CONCLUSION AND FUTURE WORK

In the MICA project we were confronted with a set of distributed sensors and actuators as well as a distributed set of modelling and controlling components, In this paper, we described our approach with a specific view on the first two layers of the adaptation process and the implementation of the inter-component communication.

At the current stage of the project we developed an architecture for distributed multi-modal applications, implemented the components on the sensoring, modelling and actuating layers. As a backend of the system, we defined a communication approach based on distributed brokering components and implemented it falling back on a chat-client delivering well-structured information exchange. The dialogue-management is currently rudimental implemented. In the next phases, we will focus to enhance the current system in order to implement more intelligent components on the controlling level.

5 ACKNOWLEDGMENTS

This work was supported by SAP Research, Contract Nr. DE-2004-044. We thank our colleagues who collaborated in the project, particularly Stefan Apelt, Markus Eisenhauer, Oliver Kaufmann, and Alexander Schneider for their fruitful ideas and contributions in the context-adaptive project.

REFERENCES

[1] Cohen, P., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. Quickste: Multimodal interaction for distributed applications. Proceedings of the Fifth ACM International Multimedia Conference, New York, NY: ACM Press, 1997, 31-40.

[2] Eccher, C., Eccer, L., Falavigna, D., Nardelli, L., Orlandi, M., and Sboner, A. On the usage of automatic voice recognition in a web based medical application. Proceedings of ICASSP 2003, Hong Kong, China, 2003.

[3] Eisenhauer, M., Lorenz, A., Zimmermann, A., Duong, T., and James, F. Interaction by movement - one giant leap for natural interaction in mobile guides. In Proceedings of the 4th Workshop on HCI in Mobile Guides, 2005

[4] Kaufmann, O. Erkennung auffälliger Verhaltensmuster zur Generierung von kontextsensitiven und proaktiven Hilfeangeboten, master-thesis, University of Koblenz, 2005

[5] Lorenz, A. A specification for agent-based distributed user modelling in ubiquitous computing. In Dolog, P. and Vassileva, J., editors, Proceedings of the 1st Workshop on Decentralized, Agent Based and Social Approaches to User Modelling, pages 31–40, 2005.

[6] Lorenz, A., Schmitt, C., Oppermann, R., Eisenhauer, M., and Zimmermann, A.. Location and tracking in mobile guides. In Proceedings of the 4th Workshop on HCI in Mobile Guides, 2005

[7] Oviatt, S., and Cohen, P. What comes naturally. Communications of the ACM, 43(3):45-53, 2000.

[8] Paterno F. (Ed.) (2002): Mobile HCI 2002, LNCS 2411. Springer-Verlag Berlin Heidelberg 2002.

[9] W3C Multimodal Interaction Framework, 2002, http://www.w3.org/TR/2002/NOTE-mmi-framework-20021202/

[10] Zimmermann, A. and Lorenz, A. Creating Audio-Augmented Environments. Journal of Pervasive Computing and Communication, 1(1): 31-42, 2005

[11] Zimmermann, A., Lorenz, A., and Specht, M. Personalization and context management. User Modeling and User Adaptive Interaction, Special Issue on User Modeling in Ubiquitous Computing, 15:275–302., 2005.