UbisWorld 3.0: a semantic Tool Set for Ubiquitous User Modeling

Dominikus Heckmann, Matthias Loskyll, Rafael Math,
Pascal Recktenwald, and Christoph Stahl

DFKI GmbH, 66123 Saarbrücken, Germany, heckmann@dfki.de

Abstract. Ubiquitous User Modeling looks at personalization within smart environments, tries to resolve problems of diversity of user modeling techniques and representations by mediation of user models, and bases on semantic standardization of user modeling enabling user modeling data exchange and sharing by using a common user modeling ontology and language. This paper describes the initial work done in the UbisWorld 3.0 project, a tool set of semantic web and Web 2.0-like services with the distributed management of scrutable user models, as well as the foundations for a distributed user model ontology development. This demonstration description previews only selected aspects of the new data types, implementation issues and the user interfaces. The complete system can be tested online as registered user at www.ubisworld.org.

Key words: ubiquitous user modeling, scrutable user model interface, user model broker, ontology editor, semantic search

1 Introduction

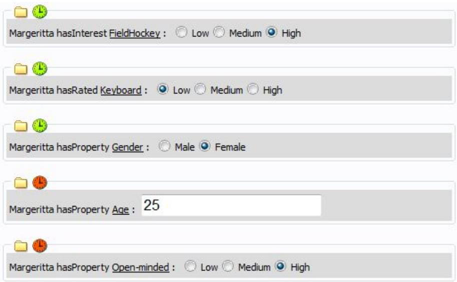
The UbisWorld project has started about seven years ago at the German Research Center for Artificial Intelligence with the goal to analyze the relation between the intersection of user modeling and ubiquitous computing. The latest tool set version UbisWorld 3.0 integrates the new era of Web 2.0 and is based on Semantic Web technologies. Mobile and ubiquitous computing raise new opportunities for assistance systems. Based on the knowledge of the user's behavior, so-called intelligent or instrumented environments are able to foresee when user assistance is needed, and the system can pro-actively offer assistance in the sense of an agent. However the design of such intelligent assistance systems requires a very precise model of the users needs in order to implement useful features and to avoid undesired or disturbing system behavior. It also requires a whole set of tools and services of which some are discussed in this demonstration description.

2 Selected Modules of the UbisWorld 3.0 Tool Set

We briefly discuss several subsystems like UbisEditor, UbisSearch, UbisPhotos, that are all implemented to analyze ubiquitous user modeling in the era of Web 2.0 and semantic web.

2.1 UbisBroker: a user model service with an introspection tool

A user model *service* manages information about users and contributes additional benefit compared to a user model *server*. The UbisBroker user model service consists of a set of application-independent servers with a distributed approach for accessing and storing user information, the possibility to exchange and understand data between different applications, as well as adding privacy and transparency to the statements about the user. The key feature is that the semantics for all concepts is mapped to the Gumo ontology and the syntax is defined in UserML. All distributed user models can be introspected and changed by a user interface as shown below.



Margeritta hasInterest FieldHockey : ○ Low ○ Medium ● High

Margeritta hasRated Keyboard : ● Low ○ Medium ○ High

Margeritta hasProperty Gender : ○ Male ● Female

Margeritta hasProperty Age : 25

Margeritta hasProperty Open-minded : ○ Low ○ Medium ● High

Fig. 1. UbisBroker with user model introspection

2.2 UbisPointer: a new data structure for semantic web and web 2.0

We invented a new mechanism to create unique identifiers for semantic web and ubiquitous computing, which we call UbisPointer. UbisPointer combine UbisPrefix and UbisToken with "..", the so called lying colon operator. If no default prefix is defined, "UBIN" is used, which stands for UBis Identification Number. The nested RelationPointer can be used to identify roles in our ontology, for instance.

```
[D]   UbisToken   ::=   ((Digit | Letter | UChar) | ("-" (Digit | Letter | UChar)))*
[D]   UbisPrefix  ::=   Upper ((Digit | Upper) | ("-" (Digit | Upper)))*
[D]   UbisPointer     ::=  DefaultPointer | PrefixPointer | RelationPointer
[D]   DefaultPointer  ::=  UbisToken
[D]   PrefixPointer   ::=  UbisPrefix ".." UbisToken
[D]   RelationPointer ::=  UbisPointer ("..." UbisPointer)+
```

UbisLabel is our approach to define multilingual labels in identifiers for semantic web and ubiquitous computing. The basic idea is to use only ASCII characters in names, even if selected special characters are allowed. The syntax of the UbisLabel approach is described below.

```
[D]   UbisLabel      ::=  SimpleLabel | LanguageLabel | MultiLabel
[D]   SimpleLabel    ::=  UbisToken
[D]   LanguageLabel  ::=  (LanguageCode)+ ".." UbisToken
[D]   MultiLabel     ::=  UbisLabel ("..." UbisLabel)+
[D]   LanguageCode   ::=  ISO_639_1 | "ME"
[D]   ISO_639_1      ::=  "EN" | "DE" | "JP" | "YI" | "ZH" | ...
```

So this approach not only enables multiple labels for any concept from the knowledge representation point of view, but also allows the adaptive selection and presentation of a label according to a situation-aware strategy.

2.3 UbisOntology: including semantic user model dimensions

The knowledge base of the system is built out of two ontologies: GUMO (General User Modeling Ontology) and UbisOntology [1]. The user interfaces to these ontologies are represented as foldable trees whose nodes consist of classes and instances. In addition, several external ontologies and taxonomies have been integrated into the UbisWorld. All the trees can be browsed in a very efficient way by using Ajax technologies [2]. In the UbisEarth tree (our largest tree with about 28 million nodes) we are already able to display roles and properties as tree nodes, too.

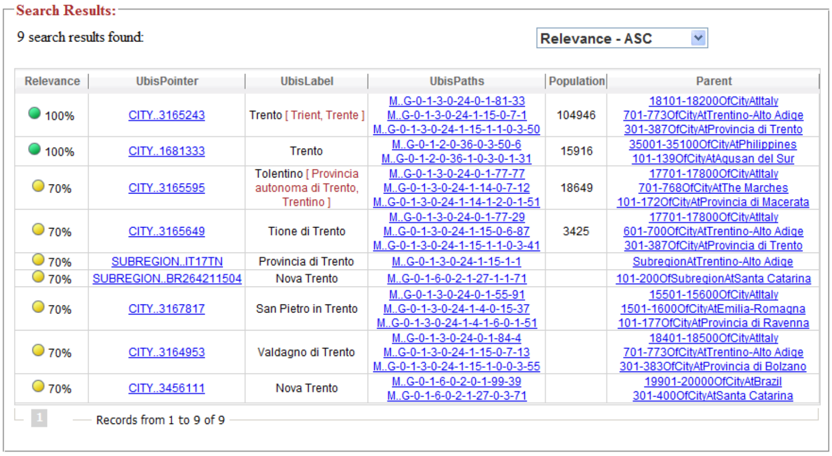
2.4 UbisSearch: adaptive search results of semantic data

UbisSearch is our search engine, which can be used to find classes, instances and properties of the UbisOntology. This is an important feature, especially when a user plans to insert a new element into the UbisWorld ontology (see 2.5), but is not sure whether such a concept already exists. We built our own inverted index, in which we assign each term that occurs in a UbisLabel of our ontologies to the corresponding list of UbisPointers. By doing so, we are able to perform search queries in an efficient way. However, the big difference compared with the Google search engine, for example, is that we search over a set of labels instead of documents.

There are three types of search results: full, partial and similar word matches. The latter ones are obtained by replacing (at this point of time only German) special characters by the appropriate ASCII characters, which is especially useful when searching for spatial concepts because in our data source several cities are labeled with the English translation of the proper name.

We display the results by using a JavaScript Grid, which supports on-demand-loading by using Ajax technologies and paging, i.e. we can present very large search result sets. The grid can be sorted either by relevance, UbisPointer, UbisLabel or population (in case of a spatial data). By default, the information is sorted firstly by relevance and secondly by population (or UbisLabel in case of non-spatial data). Figure 2 shows the search results for the key word "Trento". You can easily find the Italian city by looking at the parent column, which shows the different occurrences of this city in our ontology. One of the parents tells us that this is a city in Italy. By clicking on the UbisPointer, further information to the selected object (e.g. the selected location shown in Google Maps) is displayed at the right side of the web page. Selecting one of the UbisPaths, on the other hand, opens the corresponding node in one of our ontology trees.

These adaptable search results form a pre-step towards adaptive search in semantic representations.



Search Results:

9 search results found:

Relevance	UbisPointer	UbisLabel	UbisPaths	Population	Parent
● 100%	CITY..3165243	Trento [Trient, Trente]	M..G-0-1-3-0-24-0-1-81-33 M..G-0-1-3-0-24-1-15-0-7-1 M..G-0-1-3-0-24-1-15-1-1-0-3-50	104946	18101-18200OfCityAtItaly 701-773OfCityAtTrentino-Alto Adige 301-387OfCityAtProvincia di Trento
● 100%	CITY..1681333	Trento	M..G-0-1-2-0-36-0-3-50-6 M..G-0-1-2-0-36-1-0-3-0-1-31	15916	35001-35100OfCityAtPhilippines 101-139OfCityAtAgusan del Sur
● 70%	CITY..3165595	Tolentino [Provincia autonoma di Trento, Trentino]	M..G-0-1-3-0-24-0-1-77-77 M..G-0-1-3-0-24-1-14-0-7-12 M..G-0-1-3-0-24-1-14-1-2-0-1-51	18649	17701-17800OfCityAtItaly 701-768OfCityAtThe Marches 101-172OfCityAtProvincia di Macerata
● 70%	CITY..3165649	Tione di Trento	M..G-0-1-3-0-24-0-1-77-29 M..G-0-1-3-0-24-1-15-0-6-87 M..G-0-1-3-0-24-1-15-1-1-0-3-41	3425	17701-17800OfCityAtItaly 601-700OfCityAtTrentino-Alto Adige 301-387OfCityAtProvincia di Trento
● 70%	SUBREGION..IT17TN	Provincia di Trento	M..G-0-1-3-0-24-1-15-1-1		SubregionAtTrentino-Alto Adige
● 70%	SUBREGION..BR264211504	Nova Trento	M..G-0-1-6-0-2-1-27-1-1-71		101-200OfSubregionAtSanta Catarina
● 70%	CITY..3167817	San Pietro in Trento	M..G-0-1-3-0-24-0-1-55-91 M..G-0-1-3-0-24-1-4-0-15-37 M..G-0-1-3-0-24-1-4-1-6-0-1-51		15501-15600OfCityAtItaly 1501-1600OfCityAtEmilia-Romagna 101-177OfCityAtProvincia di Ravenna
● 70%	CITY..3164953	Valdagno di Trento	M..G-0-1-3-0-24-0-1-84-4 M..G-0-1-3-0-24-1-15-0-7-13 M..G-0-1-3-0-24-1-15-1-0-0-3-55		18401-18500OfCityAtItaly 701-773OfCityAtTrentino-Alto Adige 301-383OfCityAtProvincia di Bolzano
● 70%	CITY..3456111	Nova Trento	M..G-0-1-6-0-2-0-1-99-39 M..G-0-1-6-0-2-1-27-0-3-71		19901-20000OfCityAtBrazil 301-400OfCityAtSanta Catarina

Records from 1 to 9 of 9

Fig. 2. Adaptable Search results for "Trento"

2.5 UbisEditor: distributed user model ontology editing

UbisEditor is an online ontology editor, which shall make an end-user friendly collective knowledge building possible. The editor already supports the most important functionalities for editing an ontology: creating new classes or instances, renaming and deleting objects. By using a JavaScript context menu, we provide an efficient and easy-to-use way of performing these editor actions. Right mouse button click opens the context menu and highlights the selected node in the tree. This context menu is adaptive to the rights of the user, which depend on the user's role. This means that only users belonging to a certain role can delete objects, for example.

In addition to the simple editing of a label, we also provide a feature for multilingual label editing. When selecting a node of an ontology tree, a grid appears on the right-hand side of the UbisWorld web page. Further translations of the concept's name can be added by first selecting a language out of a combo box containing the different iso 639 language codes and entering the appropriate name into a text editor field afterwards. This functionality facilitates a collective generation of multilingual ontology concepts.

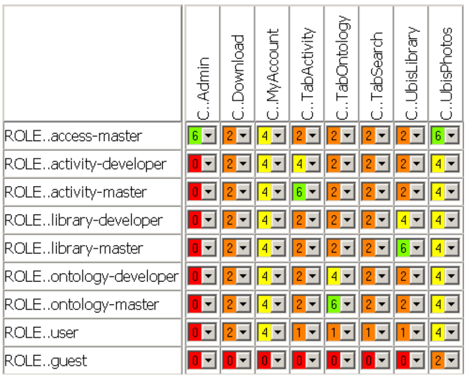
Another important feature supported by UbisEditor is the possibility to create ontology views, i.e. to extract only those parts of the ontology that are important for the scenario on hand. In editor-mode, the ontology trees are displayed with a check box on the left-hand side of each node. So the user can easily select the needed concepts and create a personalized ontology view by pushing the appropriate button. Then an OWL-file is provided for download.

2.6 UbisPhotos: a web 2.0 test bed for user modeling 2.0 issues

UbisPhotos is an approach to social software, where every user can create individual albums, upload photos and share them with other users. Unlike members of some other great photo communities, members of UbisWorld do not need to decide whether a photo is just private or public, but they can allow a pre-defined group of users to watch it. To facilitate the management of these groups, UbisGroups was introduced. By the help of this component, a user can create own groups and administrate them. There are currently three statuses a group can have: open for everybody, users need to request for membership and users can join by entering a password obtained by the creator. In the foreseeable future there will also be a possibility to invite users to a group. When somebody joines a group or is still waiting for approval, the group owner will be notified to eventually approve or reject the membership. From a user's personal list of groups he joined or created, the user can pick any groups whose members he wants to grant access to his photos. Further options to customize one's personal photo album are rearranging photos on a drag-and-drop basis, renaming, deleting, moving photos to other albums and writing comments based on Ajax-requests. Apart from the possibility to share photos and albums with members of pre-defined groups, a user can at any time share the 22 characters long base62 identifier of an individual album or photo, which grants the receiver permanent access even if a photo is marked as private.

2.7 UbisAccess: privacy and security aspects in distributed UM

To provide a personalized environment across all components of the UbisWorld, a profound rights management has to be implemented and applied. We group the users into different roles. Each role is a description of rights, which are granted to a user belonging to a particular role. Since a user has to be identified, to grant him access to his personal content, the user must login in the first place with the user name and password he selected during registration. After successful registration, the new member is assigned to the role "ROLE..user", which allows creation of a user profile and photo albums to share with other users. Furthermore by this role almost every content of the UbisWorld can be viewed, however it does not permit any write access to contents that were created by other users. For these purposes, other user roles as "ROLE..ontology-developer", "ROLE..library-developer", "ROLE..activity-developer", etc. are needed to create new or edit existing ontologies, book information or activity scenarios, respectively. Even more rights are related to user roles as "ROLE..ontology-master", "ROLE..library-master", "ROLE..activity-master", etc. whereas affiliated users may also delete contents belonging to their scope of responsibility. The top level in the hierarchy of roles constitutes "ROLE..access-master", which promotes standard users to administrators deciding on other users' rights. Since user roles just cover single components (UbisOntology, UbisLibrary, UbisActivity, ...) of the UbisWorld, the adapted environment for each user can be arranged by assigning multiple roles to one user. If a user's right to access a specific content can be obtained by two roles and both provide different values, the higher right will be the decisive factor. The connection between user roles and content can be seen in figure 3.



	C..Admin	C..Download	C..MyAccount	C..TabActivity	C..TabOntology	C..TabSearch	C..UbisLibrary	C..UbisPhotos
ROLE..access-master	6	2	4	2	2	2	2	6
ROLE..activity-developer	0	2	4	4	2	2	2	4
ROLE..activity-master	0	2	4	6	2	2	2	4
ROLE..library-developer	0	2	4	2	2	2	4	4
ROLE..library-master	0	2	4	2	2	2	6	4
ROLE..ontology-developer	0	2	4	2	4	2	2	4
ROLE..ontology-master	0	2	4	2	6	2	2	4
ROLE..user	0	2	4	1	1	1	1	4
ROLE..guest	0	0	0	0	0	0	0	2

Fig. 3. Table of Rights - Rights are abbreviated by digits: 0 - no access; 1,2 - read access; 3,4 - write/edit access; 5,6 - delete access

Only users belonging to "ROLE..access-master" may see this table and change the settings of rights between user roles and content (marked in table with prefix C). The greater the digit connecting a role with a content, the higher are the access rights of users belonging to a role. The highest right is delete (digit 6). Apart from administrators deciding to what roles a user belongs to, the user himself can temporarily switch his roles to inactive by selecting them in his profile. Thus a library master is able to view the changes he made by the eyes of a standard user.

3 Conclusions and Future Work

We have presented a selected set of issues of the UbisWorld 3.0 project. If some of these modules seem not closely related to the research field of user modeling, adaptation and personalization on its own, the integrated view leads to an interesting test bed for research on ubiquitous user modeling in the era of Semantic Web and Web 2.0 facing the Internet of Things in ubiquitous computing. The UbisWorld 3.0 tool set can be tested online at www.ubisworld.org. Its initial research results are currently transferred to the domain of distributed technology enhanced learning in the EU project GRAPPLE. As future work we plan evaluations and improvements of the user interfaces as well as the data structures.

References

1. Heckmann, D.: Ubiquitous User Modeling. Berlin: Akademische Verlagsgesellschaft Aka GmbH (2006)
2. Loskyll, M.: Ontological and Ajax-based Extension of UbisWorld. Chair for Artifical Intelligence, Prof. Dr. Dr. h.c. mult. Wahlster, Saarland University (2007)