# Addressing Challenges of Ubiquitous User Modeling: Between Mediation and Semantic Integration

Shlomo Berkovsky[1], Dominikus Heckmann[2], and Tsvi Kuflik[3]

[1] CSIRO – Tasmanian ICT Centre, Hobart, Australia
shlomo.berkovsky@csiro.au
[2] DFKI GmbH, Saarbruecken, Germany
heckmann@dfki.de
[3] The University of Haifa, Mount Carmel, Haifa, Israel
tsvikak@is.haifa.ac.il

**Abstract.** Ubiquitous User Modeling aims at providing personalized services to inhabitants of smart environments. Current research in ubiquitous user modeling focuses on two directions. The first is a practical approach that tries to resolve current problems of sparseness of data and heterogeneity of user modeling techniques and representations by mediation of user models or building hybrid systems. The second approach is based on semantic standardization of user modeling enabling user modeling data exchange and sharing by using a common user modeling ontology and language. Although both approaches have their limitations, their integration has the potential to leverage their advantages and overcome the limitations. This paper discusses initial work done in this direction, suggests a path for such integration, and points out research directions aimed at bridging the gap between these approaches.

## 1 Introduction

Ubiquitous user modeling aims at providing personalized services to inhabitants of smart environments. Interest in ubiquitous user modeling is growing rapidly, mainly due to the fact that mobile and pervasive computers are widely spread and their users may benefit from personalized "information-on-the-go" services. To provide personalized services, there is a need for knowledge about the specific user, the application domain of the service, and the specific context in which the service will be provided to the user. For example, consider a user visiting an ethnographical museum exhibition with his/her family. For the provision of personalized information services, gastronomical preferences of the visitor seem irrelevant, whereas his/her historical knowledge is relevant. The artistic properties of the presented cave paintings may not be relevant to the average visitor, whereas they may be very relevant if art is the main interest of the visitor. Turning to context, although the visitor may be a knowledgeable art expert, the context of a family visit may affect the provided service, such that the delivered information will use lay terms rather than more appropriate for the visitor artistic knowledge.

In general, some user characteristics, (such as preferences for instance), represented by a user model may be valid only within specific contextual conditions, such as spatial, temporal, emotional, and other conditions. That is, a user's preferences stored in

T. Kuflik et al. (Eds.): Advances in Ubiquitous User Modelling, LNCS 5830, pp. 1–19, 2009.

the user model may change as a function of various contextual conditions. The challenges of such user modeling data representation were discussed and exemplified by the multi-dimensional *experience* model suggested by Berkovsky et al. [2006; 2008], which extended traditional two-dimensional recommender systems approach addressing users and items only [Adomavicius et al., 2005]. When attempting to personalize a service provided to a user, there are several aspects that should be taken into account: 1) personal characteristics of the user requesting the service, 2) characteristics of the service itself, and 3) contextual aspects. In the context of recommender systems, [Berkovsky et al., 2008] defined experience as a function that maps the tuple {*user* that had the experience, *item* experienced, *context* of the experience}, to the *evaluation* of the experience. Formally, the experience was represented by:

$$Exp: User_{feat} \times Item_{feat} \times Context_{feat} \rightarrow evaluation.$$

$User_{feat}$, $Item_{feat}$ and $Context_{feat}$ refer to the representations of, respectively, the user features, item features and context features, while *evaluation* represents the feedback provided. Figure 1 schematically illustrates the above three-dimensional representation of experiences.

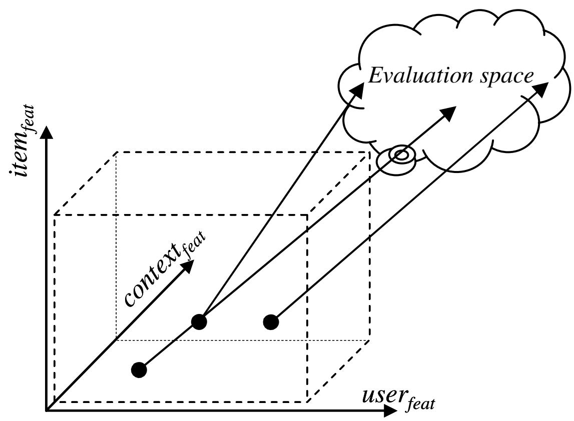


**Fig. 1.** Representation of Context-Aware Experiences in Three-Dimensional Space

The specific dimentions, $User_{feat}$, $Item_{feat}$ and $Context_{feat}$, in turn, may be described using a multidimensional representation by sets of features. Hence, a user model, a domain description, and contextual conditions of the experiences are referred to as subspaces of this three-dimensional space. For instance, in the above mentioned family museum visit example, only one contextual feature of *companion* out of a large $Context_{feat}$ set of contextual features is used. When the *companion* feature is assigned the value of *colleagues*, the evaluation of the delivered information using the artistic terminology may be positive, while when the *companion* features has the value *family*, the evaluation may be negative.

One of the most challenging questions in this setting is the initialization of the user models. In other words, how can the system provide an accurate personalized service to a user on his/her first interaction with the system, when none or little information about him/her is available to the system? In order to do that, the system needs to acquire some contextualized information about the user (while application domain characteristics may be available). Traditionally, personalization services were initialized by explicitly providing personal information or rating sample items. However, this is impractical or time- and effort-consuming in a ubiquitous computing environment. Hence, there is a need for a fundamentally different approach, where the user model initialization task is rather based on interoperability of personalization systems, i.e., deriving the missing information from information previously acquired by other systems, or possibly the user's personal devices. This implies bridging between the differences and discrepancies in terminologies, concepts, and user modeling approaches used by various systems, since nowadays personalization systems are typically designed to deliver their own personalized service, making user modeling information sharing practically impossible.

Recent research outlines two major approaches for such interoperability. The first calls for using a generally agreed, standard user modeling ontology, as suggested by Heckmann [2005], while the second addresses the practical limitations of personalization services by suggesting the idea of mediation [Berkovsky et al., 2008]. Heckmann [2005] suggested a rich and standardized ontology for user modeling, augmented by XML-based a user modeling language for information sharing [Heckmann and Krueger, 2003]. The benefits of this approach are clear: the agreed upon ontology and standardized XML-based representation pave the way for user modeling information sharing. An obvious limitation of this approach is that it requires all systems to adhere to the standard user model ontology, which brings up the question whether service providers will accept this requirement. Berkovsky et al. [2008] idea of mediation addresses the challenges of user modeling information sharing across applications in a practical way. Mediation deals with transferring user modeling data from one representation (for example, collaborative filtering) to another (for example, content based filtering) in the same domain, or across domains. Although the mediation does not imply standardized ontology, practical mediation scenarios require a large number of transfer mechanisms to be developed.

Both approaches have various variants implemented, demonstrated, and evaluated. However, both approaches could benefit from bridging the gaps between them and integrating components of one into another. Such an integrated user modeling data interoperability framework will enable developers of personalized services to use the level of abstraction and generality that best suits their case. This paper suggests integrating both approaches, and points out a research agenda for bridging that gap between them, while demonstrating initial steps already taken in this direction. The rest of the paper is organized as follows: Section 2 provides background and related work; Section 3 provides a description of initial step towards bridging the gap by smart situation retrieval with semantic conflict resolution; Section 4 concludes with suggestions for future work required to further bridge the gap between the two extremes.

## 2 Background and Related Work

Integration and reuse of user modeling mechanisms and data are drawing research interest for more than a decade. Various approaches were explored over time and it seems that two orthogonal approaches have evolved: (1) a comprehensive user modeling ontology that strives to provide rich semantics for standardization of user modeling, and (2) user modeling mediation, aimed to resolve the practical problems of heterogeneity in both user modeling representation and user modeling techniques. Both approaches are overviewed below, as well as additional related work.

### 2.1 Generic User Models and User Modeling Servers

User model initialization is a well known problem in personalization. Over the years, various approaches have been suggested to address it and shorten the process. As surveyed by Kobsa [2001], pioneering work of generic user models started as early as the mid-1980s, with the intention to allow re-use of already developed user modeling components and systems, thus focusing on technology re-use, rather than on the re-use of precious user modeling data collected in practice. It could be understood at that time, when in general, applications were stand alone and specific and user modeling capabilities were integrated into the application. The first step into ubiquitous user modeling was made by decoupling the linkage between the application and the user modeling component and introducing the general user modeling shell systems [Kobsa 2001]. Such shells, servers and toolkits were developed starting at the early 1990s. Kobsa [1995] performed a brief domain analysis of generic user modeling shells and listed the common core of services. This was later on extended by Kobsa [2001] that defined more abstract requirements. However, until this point the focus was the system – generic mechanisms for user modeling that could potentially be applied in different domain applications "as is", as needed, provided that the relevant domain knowledge and users' personal data are available. During the late 1990s, commercial user modeling shell systems started to appear, applying client-server architecture. This architecture provided the initial step towards sharing and re-using user modeling data for personalization by different applications [Pazani 2000, Kobsa 2001].

Kobsa [2001] also brought up the need to import and export existing user data as a requirement from user modeling server, but without suggesting any mechanism or framework for that process. He states correctly that processing done by current servers cannot be used outside the context of the specific domain and application due to the lack of abstract representation of learned users' characteristics. [Kobsa 2001] also details the requirements that will facilitate wide dissemination of generic user models. Originally, the requirements were split between academic and commercial applications, but since both groups of requirements were complimentary, they are integrated below into one list (omitting the technical performance requirements):

- Generality – domain independence, compatibility with as many as possible applications and domains, and for as many as possible user modeling tasks.
- Expressiveness – ability to express as many as possible types of facts and rules about the user.
- Inferential capabilities – capability of performing various types of reasoning and resolving the conflicts when contradictory facts or rules are detected.
- Import of external data – ability to integrate the user modeling data collected by the system with the data collected by other systems.
- Privacy – support of privacy policies and conventions, national and international privacy legislations, and privacy-supporting tools and service providers.

- Quick adaptation – ability to quickly adapt services to new users, personalization functionalities, applications, and domains.
- Extensibility – provide application programmer interfaces (APIs) and interfaces that allow (possibly bi-directional) exchange of user information between user-modeling tools, thus allowing the integration of variety of user modeling techniques.

Kobsa [2001] concludes his survey of generic user modeling systems with fairly accurate predictions of the evolvement of networked computers and especially mobile computing. He suggests two options for ubiquitous user modeling with a user model residing on the server side or on the client side, e.g., on the mobile device carried by the user. Furthermore, he presents the issue of personalization of smart appliances and the potential of multiple-purpose usage of users characteristics and discusses in light of this the pros and cons of client side versus server side user models [Yimam and Kobsa, 2003]. The survey is summarized with "*…one can expect to find a wide variety of generic user modeling systems, each of which is going to support only a few of the very different future manifestations of personalization and other applications of information about the user*". The conclusion from the above, (on the one hand, the expected variety of limited user modeling servers, and, on the other hand, the usefulness of re-using already available precious user modeling data), brings forward the need for some kind of generic mechanism for user modeling data sharing, conversion and exchange.

Recently, Van der Sluijs and Houben [2005] introduced Web 2.0 technology into user modeling servers when they introduced GUC – a Generic User Modeling Component. They suggest a user modeling server using OWL for user models representations stored in user models repository and applying schema matching techniques for finding appropriate user models in the repository as a response to a service request. This is in fact a suggestion how to apply novel web 2.0 technology for the above described user modeling servers' idea.

### 2.2 Semantically Enhanced User Modeling

Standardization and "common language" is one of the key issues in integrating information sources in every domain, including user modeling. The state-of-the-art approach for the problem of standardization of domain-specific knowledge representation is the use of ontologies. According to Gruber [1993], ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. These concepts constitute the domain vocabulary, whereas the relationships link them into a meaningful domain structure. Ontologies and common language communication protocols are among the commonly expected approaches, while the advent of the semantic web provided a common platform that encourages and supports this approach.

Ontology-based representation of user modeling was discussed by Kay [1999], which motivated ontology-based reusable and scrutable, i.e., understandable, modeling of students. Reusability allowed separating the representation of the user model from the personalization task in a particular application or domain. The structure of the user models was based on a set of predefined and agreed upon ontologies facilitated access to a customized explanation of the meaning of the user modeling components in each domain. However, despite the great potential in the use of ontologies, they did not become widely used in user modeling tasks, possibly due to the considerable initial effort required in the construction of any ontology.

The notion of generic ontology-based user models was first developed by Razmerita et al. [2003] that presented a generic ontology-based user modeling architecture called OntobUM. OntobUM integrated three ontologies: user ontology characterizing the users, domain ontology defining the relationships between the personalization applications, and log ontology defining the semantics of user-application interaction. Mehta et al. [2005] and Mehta and Nejdl [2007] also suggested the use of ontology for standardization of user models and to ease information exchange between applications. A similar, but way more extensive approach for ontology-based representation of the user models was presented by Heckmann et al. [2006]. Heckmann [2006] suggests GUMO[1] that seems to be the most comprehensive publicly available user modeling ontology to date. Vassileva et al. [2003] also noted the need for a standard catalogue for user modeling, which defines relevant parameter values, and mechanisms for different user modeling purposes, as a necessary tool for integrating user models fragments.

The above works are natural extensions of earlier works on general user modeling systems of [Kay, 1995], [Kobsa, 2001], [Jameson, 2001], and others. Such ontology may be represented in a modern semantic web language like OWL, and thus be available for all user-adaptive systems at the same time. The major advantage of such approach would be the simplification of exchanging user model data between different user-adaptive systems. Even though this is a desired situation and GUMO seems to be a major step in enabling the achievement of such a goal, the current state of the art is different. So far, there is a great deal of syntactical and structural differences between existing user modeling systems that cannot be overcome simply be introducing a commonly accepted taxonomy, adapted to user modeling tasks as suggested by Heckmann, [2006]. In addition to GUMO, the UbisWorld[2] knowledge-base has been designed to complement GUMO and model contextual characteristics of a user, including their activity, as well as the environmental context. It also provides a symbolic spatial model to express location. Heckmann [2006] acknowledges the need for a relevant domain-specific ontology, as part of the overall framework, but rightfully recognizes the problem of including such ontologies in a user model. Heckman's compromise is to include a general interest model in the user model, a solution that needs to be extended for specific applications (by adding domain-specific ontology) in order to allow the application of GUMO in every specific domain.

### 2.3 User Models Integration and Mediation

Vassileva et al. [1999; 2001; 2003] pointed out the future situation of fragmented and inconsistent user models in ubiquitous computing. They suggested a distributed Multi-Agent approach for addressing the challenges of ubiquitous computing where large number of inconsistent user model fragments may be available and there will be a need to integrate them for an ad-hoc personalized service delivery. They presented

[1] GUMO homepage: `http://www.gumo.org`
[2] UbisWorld homepage: `http://www.ubisworld.org`

I-Help [Vassileva et al., 2003], a system providing access to help resources for students. The user modeling information included preferences, rankings, ratings and numeric overlays on course topics. The system is based on matchmaking – a variety of broker agents that keep track of user models and are able to map help requests to possible service providers. This matchmaking is based on domain taxonomy (provided by an instructor/teacher in I-Help). In their view, user modeling is a process that involves computation over subjects, objects, purposes and resources, where the I-Help is a specific demonstration. In order to generalize the approach, they noted the need for a "catalogue of purposes for user modeling" that needs to be manually constructed and be a standard reference to user modeling.

While Vassileva et al. [1999; 2001; 2003] and McCalla et al. [2000] focused the discussion on an abstract user modeling process, Berkovsky et al. [2008], suggested a practical approach aimed at overcoming the sparseness problem in recommender systems, by using *mediation* of the user models and other user modeling data. The exact definition of mediation is formulated as follows: "*mediation of user models is a process of importing the user modeling data collected by other (remote) recommender systems, integrating them and generating an integrated user model for a specific goal within a specific context.*" In this definition, the term integration refers to a set of techniques aimed at resolving the heterogeneities and inconsistencies in the obtained data. The mediation process facilitates the instantiation of user models by inferring the required user modeling data from past experiences and their evaluations in a three-dimensional context-aware representation space. Hence, the mediation enriches the existing (or bootstraps empty) user models in a target recommender system using the data collected by remote systems. This, in turn, facilitates provision of better context-aware recommendations.

The main obstacle for materializing the mediation ideas is overcoming the heterogeneity of the user modeling data. For example, recommender systems from different application domains imply different user modeling data stored in the models. Within the same domain, different systems may store different information in their user models, according to the specific recommendation technique being exploited (e.g., collaborative filtering ratings [Herlocker et al. 1999] versus domain/item features in content-based systems [Morita and Shinoda 1994]). Moreover, even the models of two recommender systems from the same application domain exploiting the same recommendation technique may use different terms to describe equivalent underlying objects, i.e., users, items, or domain features. Hence, successful completion of the user model mediation task requires (1) developing and applying reasoning and inference mechanisms for converting user modeling data between various representations, applications and domains, and (2) exploiting semantically-enhanced knowledge bases, actually facilitating the above reasoning and inference.

In the domain of recommender systems, prior research tried to integrate multiple recommendation techniques in the recommendation generation process. These systems are referred to in the literature as hybrid recommender systems [Burke 2002]. Although hybrid recommenders typically combine several recommendation techniques into a single recommender system for the sake of improving the accuracy of the generated recommendations, they are not concerned with the conversion of user modeling data between independently operating recommender systems. Hence, it should be noted that the mediation of user modeling data is more generic, dynamic and flexible approach than the data hybridization methods presented in [Burke 2002]. In a mediation scenario, the user model data a system received may be originated by various systems using different recommendation techniques and the mediation implies an ad-hoc application of dynamically selected mediation modules converting the user modeling data from a source to the target system, whereas classical hybridizations integrate specific techniques and approaches.

## 3 Smart Situation Retrieval with Semantic Conflict Resolution

The above two approaches refer to the issue of interoperability of personalization and user modeling systems in two orthogonal ways. Every approach has its own inherent limitations. Ontology-based standardization depends on a voluntary adoption of some kind of user modeling data representation standard by personalized service providers. Although GUMO currently represents a major step towards such standardization, at the current state of affairs this is a wishful thinking, since served eproviders need to adopt the standards and agree to share information. Mediation, on the other hand focuses on transforming specific models (or parts of such models) between applications, hence a mediator is needed between every two methods and user modeling data representation and terminology pose another challenge on practical mediation. The possible benefits of combining domain specific knowledge and more abstract user model knowledge were noted already both by Heckmann [2006] and by [Berkovsky et al., 2007]. The natural question we are facing is how to enhance the mediation mechanisms with semantic knowledge, in a way that will allow gradual adoption of standard tools like GUMO and UserML, while allowing the continuous use of the specific user modeling techniques applied in specific applications. In other words, how can an application be enhanced without the need to completely replace personalization mechanisms?

The *SmartSituationRetrieval* [Heckmann and Blass, 2008], is an example of a step towards this direction of semantic abstraction of user modeling for personalization. In this specific case, semantic abstraction is used for contextual conflicts resolution process. One class of problems that may occur in the challenge of context integration is the problem of *semantic conflicts* that occur in a case where several context statements use different words, concepts, ranges or values to describe the same situation. For example if one system claims that “Peter is happy,” and the other system says “Peter is not happy,” it is a classical conflict that has to be detected and resolved (which is reasonably easy in this case). On the contrary, if the other system says “Peter is sad,” the system has to understand the semantic relation between `happiness` and `sadness` to detect these two statements as being conflicting. Consider another example, where one system talks about `blood pressure` while the other talks about `pulse` and both mean the same context dimension. Finally, consider a third example, where one system says “Peter is in the `grocery store,`” while the other system only reports “Peter is in a `shop.`” The crucial point is that these contextual dimensions are semantically related. In order to handle these relations there is a need for an ontology that will cover this semantic information.

UbisWorld’s user model exchange and context management system UbisMEMORY is based on the semantic web ontology GUMO that describes the user model and context dimensions, but not the semantic relations between the different dimensions. However, these relations are defined and collected in the WordNet ontology, which will be presented in the following sub-section.

### 3.1 WordNet

The basic concept of WordNet is a collection of *SynSets*. A SynSet groups words with synonymous meaning. For example, "`heartbeat`, `pulse`, `pulsation`, `beat`" would be one SynSet of the word `heartbeat`. However, `heartbeat` is also a part of another SynSet "`heartbeat`, `flash`, `blink of an eye`, `split second`," in the meaning of *"Everything went so fast, in a heartbeat it was over."* To distinguish between the different meanings of the same word in different SynSets, one talks about *WordSenses*. Hence, SynSet contains one or more WordSenses and each WordSense belongs to exactly one SynSet. In turn, each WordSense has exactly one *Word* that represents it lexically, and one Word can be related to one or more WordSenses [Van Assen, 2002]. Figure 2 schematically presents a graphical representation of Words, WordSenses and SynSets as part of the above example. Since both RDF/OWL extension of WordNet and the general user model and context ontology GUMO are represented in RDF/OWL, the representation used for both ontologies hereafter will be in RDF/OWL.

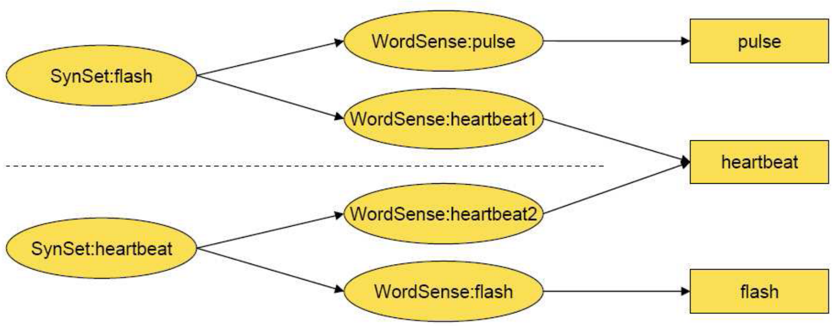


**Fig. 2.** Example of WordNet: The three words pulse, heartbeat and flash and their corresponding WordSenses and SynSets. The WordSense pulse is synonym to one of the WordSenses of heartbeat, so is flash with another WordSense of heartbeat. [Blass 2007]

As indicated by the symbols used in figure 2, there exists a model of WordNet using RDF and OWL. Every *SynSet*, every *WordSense* and every *Word* have unique identifiers and become RDF resources. Every *SynSet* holds the information which *WordSenses* are contained in it with the help of an RDF relation *containsWordSense*. The *WordSenses* point to the *Words* via *word* relation, which is related to the string literal it represents by *lexicalForm* relation. In addition to the relations *containsWordSense*, *word*, and *lexicalForm* there exist a number of additional relations. Some of the most widely used relations are briefly exemplified below. *Hyponym* describes a word or phrase whose semantic range is included within that of another word. For example: 'banana', 'apple', and 'grape' are all hyponyms of 'fruit'. In this example 'fruit' would be the *hypernym* of the other words. *Antonyms* are word pairs that are opposite in meaning, such as 'hot' and 'cold' or 'happy' and 'sad'. The *antonymOf* relation models this. *Meronym* denotes a constituent part or a member of something. That is, *{A}meronymOf{B}* if A is either a part or a member of B. For example, 'finger' is a meronym of 'hand' because a finger is part of a hand. Similarly 'wheel' is a meronym of 'automobile'. A further discussion on WordNet relations can be found in [Van Assen, 2002].

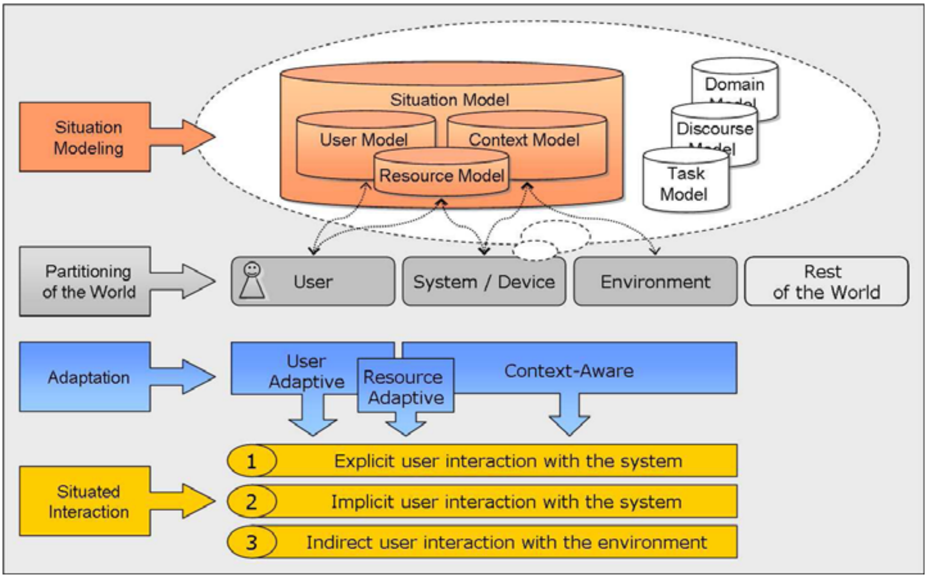


**Fig. 3.** Situated interaction and the system's situation model for mobile computing

### 3.2 Integrated Model for Context-Awareness and User-Adaptivity

The research areas of *user-adaptivity*, *context-awareness* and *ubiquitous computing* find their intersection in the concept of context, while *semantic web* technology could serve as a mediator between them. In [Kray, 2003] it is pointed out that throughout the different research communities and disciplines, there are various definitions of what exactly is contained in the *context model* [McCarthy and Buvac, 1998], the *user model* [Day, 1999], and the *situation model* [Jameson, 2001]. Therefore, it is necessary to clarify how those terms will be used in our approach. A *situation model* is defined as the combination of a *user model* and a *context model*. Figure 3 presents a diagrammatic answer to the question "*What is situated interaction and how can we conceptualize it?*" *Resource-adaptivity* overlaps with *user-adaptivity* and *context-awareness* because the human's cognitive resources fall into the user model, while the system's technical resources can be seen as part of the context model. The fundamental data structure is the SITUATIONALSTATEMENT (see [Heckmann, 2003]) that collects apart from the main contextual information also meta-data like temporal and spatial constraints, explanation components, and privacy preferences. Distributed sets of SITUATIONREPORTS form a coherent, integrated, but still hybrid accretion concept of ubiquitous situation (user and context) models.

### 3.3 User Modeling and Context Modeling with GUMO and UserML

GUMO, the general user modeling ontology allows user modeling applications to collect the user's dimensions modeled by user-adaptive systems like the *heartbeat rate*, *age*, *current position*, *birthplace, ability to swim*, and many others. The contextual dimensions like *noise level* in the environment, *battery status* of the mobile device, or the *weather* conditions are modeled as well. The main conceptual idea in SITUATIONALSTATEMENTS is the division of user model and contextual dimensions into three parts: `auxiliary`, `predicate` and `range`. Apart from these `main` attributes, there are predefined attributes about the `situation`, the `explanation`, the `privacy` and the `administration`. Thus, our basic context modeling is more flexible than simple attribute-value pairs or RDF triples. If one wants to say *something about the user's interest in football*, one could divide this into the `auxiliary`=*hasInterest*, the `predicate`=*football* and the `range`=*low-medium-high*.

GUMO is designed according to USERML approach, as an XML application (see [Heckmann and Krueger, 2003]), to facilitate easy exchange of user modeling data. Approximately one thousand groups of `auxiliaries`, `predicates` and `ranges` have so far been identified and inserted into the ontology. However, it turned out that actually everything can be a `predicate` for the `auxiliary` *hasInterest* or *hasKnowledge*, what leads to a problem if work is not modularized. The suggested solution is to identify basic user model dimensions on the one hand, while leaving the more general world knowledge open for already existing other ontologies on the other hand. Candidates are the general suggested upper merged ontology SUMO [Pease et al., 2002] and the UBISONTOLOGY[3] [Stahl and Heckmann, 2004] used to model intelligent environments. Identified user model and context `auxiliaries` are, for example, *hasKnowledge*, *hasInterest*, *hasBelief*, *hasPlan*, *hasProperty*, *hasPlan,* and *hasLocation*. A class defines a group of individuals that belong together because they share some properties. Classes can be organized in a specialization hierarchy using the *`subClassOf`* `relation`.

### 3.4 Smart Situation Retrieval Process

In an "open world assumption" together with an "open to everyone assumption", every user and every system is allowed to enter statements into repositories (that contain partial user models), where some of this information might be contradictory. Conflicts among the statements like, for example, a contradiction caused by different opinions of different creators or changed values over time, are loosely categorized in the following listing.

1. On the semantic level: the systems do not use the same ontology to represent the meaning of the concepts, which leads to the user model integration problem.

---

[3] UbisWorld homepage: `http://www.ubisworld.org`

2. On the observation and inference level: several sensors interpret the same observations differently, measurement errors occur, or systems have preferred information sources
3. On the temporal and spatial level: information is out of date or out of spatial range.
4. On the privacy and trust level: information is hidden, incomplete, secret or falsified on purpose.

The architectural diagram in Figure 4 shows the SMARTSITUATIONRETRIEVAL or in other words: how the conflict-free partial user models are generated. Not all of these modules are explained here in detail. See [Heckmann 2006] for a detailed description. The focus in our discussion is set on the semantic conflict resolution part. *Note that the oval numbers indicate the reading order.*

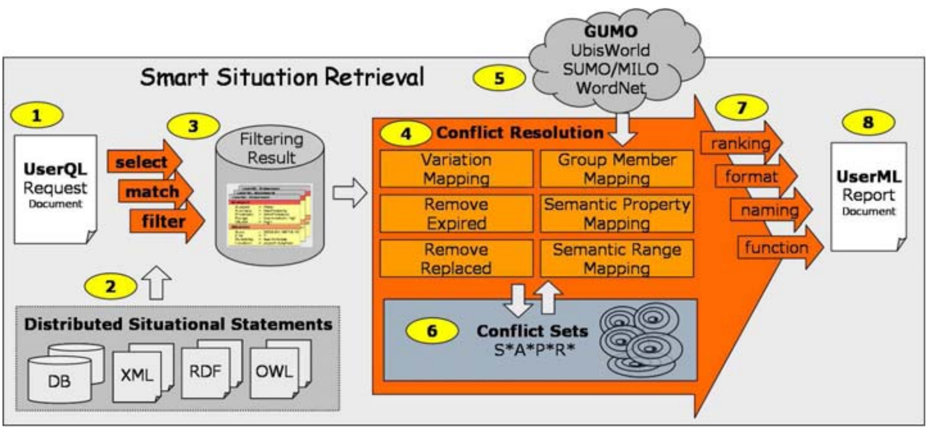


**Fig. 4.** Smart Situation Retrieval with Focus on Semantic Conflict Resolution

Item (1) shows a request that has to be parsed. It is given in UserQL, the query language that has been defined in analogy to UserML. Item (2) refers to the distributed retrieval of SITUATIONALSTATEMENTS (accumulated over time, entered by different sources) . In the retrieval case, that we are discussing here, we can see (1), (2) and (5) as given and the others as being calculated. Item (3) summarizes the three macro-steps, i.e., `select, match, and filter, and` presents the FILTERINGRESULT as input to the follow up conflict resolution process. The filtering result contains all statements that fit to the UserQL query, however with possible conflicts and contradictions.

Now, the conflict resolution phase starts. Item (4) stands for three syntactical procedures VARIATIONMAPPING, REMOVEEXPIRED and REMOVEREPLACED. These three procedures align the statements syntactically, and remove outdated and replaced statements. Item (5) represents three semantic procedures GROUPMEMBERMAPPING, SEMANTICPROPERTYMAPPING and SEMANTICRANGEMAPPING that base on the data represented by the knowledge base of WordNet and GUMO, UbisWorldOntology, and SUMO/MILO ontologies. Item (6) shows the detection of syntactic and semantic conflicts and the construction of the conflict sets. Item (7) refers to the post-processing of `ranking`, `format`, `naming` and `function` that control the output format. Item (8) forms the resulting `UserML` report, that is send back to the requestor.

A simplified example will demonstrate the process. Assume there are two statements: "Peter is 30% happy" and "Pedro is almost sad" that are stored as distributed statements in the repository (facts that were inserted sometime before) (2). Now, our system receives a UserQL request (1) like "Is Peter happy?". The select and *matching procedure (3)* compares all given match attributes with the corresponding statement attributes – in our example, the following attributes may be compared: happy and sad, Peter and Pedro (which are the only informative attributes in these statements). The *filtering procedure* operates on the matching results. Each statement is individually checked if it passes the *privacy filter*, the *confidence filter,* and the *temporal filter (in our case the temporal characteristics of the statements, whether any of them is outdated).* Now we look closer to the conflict resolution step (4), the "variation mapping" can map "Peter" in system A to "Pedro" in system B, if both denote the same person (should be reasoned about). Now, the semantic property mapping has to map "happy" and "sad" to each other, which makes sense only if there is a strong semantic relation between the two properties. The last step would be the semantic range mapping that maps "almost" onto the scale of percentage, such that it can be directly compared with "30%".

The question that arises: how do we resolve conflicts that we have found in (6) together with the defined semantics in (5). In our example, we can conclude that Peter is not happy. Independently of who has claimed which statement, like a novice versus an expert. If we also take such meta-information into account we can resolve further conflicts. C*onflict resolvers* were developed to control the *conflict resolution process* such that an ordered list of resolvers defines the *conflict resolution strategy*. These resolvers are needed if the *match process* and *filter process* leave several conflicting statements as possible answers. The *most(n)*-resolvers use meta data for their decision. Several most(n)-resolvers are presented in the following listing.

- **mostRecent(n).** If sensors send new statements on a frequent basis, values tend to change more quickly as they expire. This leads to conflicting non-expired statements. The *mostRecent(n)* resolver returns the *n* newest non-expired statements, where *n* is a natural number between 1 and the number of remaining statements.
- **mostNamed(n).** If there are many statements that claim A and only a few claim B or something else, than *n* of the most named statements are returned. Of course, it is not sure that the majority necessarily tells the truth but it could be a reasonable rule of thumb for some cases.
- **mostConfident(n).** If the confidence values of several conflicting statements can be compared with each other, it seems to be an obvious decision to return the *n* statements with the highest confidence value.
- **mostPersonal(n).** If the `creator` of the statement is the statement's `subject` (a self-reflecting statement), this statement is preferred by the *mostPersonal(n)* resolver. Furthermore, if an *is-friend-of relation* is defined, statements by friends could be preferred to statements by others.

Although these conflict resolver rules are based on common sense heuristics, they do not necessarily need to be true for specific sets of statements. An important issue to keep in mind is that the resolvers and their strategies imply uncertainty. To reflect this, the `confidence` value of the resulting statement is changed appropriately and the conflict situation is added to the `evidence` attribute.

To come back to the discussion about WordNet: the semantic conflicts are resolved using WordNet query expansion algorithm. The query expansion algorithm posts four queries to the WordNet repository: one query for synonyms, one for hyponyms, and the other two for antonyms. The two queries are shown in Figure 5. *Predicate* denotes the input of the function. It is assumed to be a string representing the full identified of the resources annotating the WordSense. All results, that is, the identifiers of related WordSenses, are added to the output.

```
SYNONYMS:
SELECT DISTINCT WS2
FROM {SynSet} wn20schema:containsWordSense
{<predicate>},
{SynSet} wn20schema:containsWordSense {WS2}
USING NAMESPACE
wn20schema =
<http://www.w3.org/2006/03/wn/wn20/schema/>

HYPONYMS:
SELECT DISTINCT WS2
FROM {SynSet} wn20schema:containsWordSense
{<predicate>},
{SynSet2} wn20schema:hyponymOf {SynSet},
{SynSet2}wn20schema:containsWordSense {WS2}
USING NAMESPACE
wn20schema =
<http://www.w3.org/2006/03/wn/wn20/schema/>";
```

**Fig. 5.** Queries for synonyms and hyponyms of a given *predicate[Blass2008]*

Statements that have predicates synonym, hyponym and antonym to the given one are found by posting queries with replaced predicates to our basic matching query engine. It could be hypothesized that synonyms, hyponyms and antonyms would improve the performance of our search, since statements with predicated synonym are semantically equivalent to the ones with the base predicate. Antonyms correspond to negation and hence if used as predicates, only the object needs to be inverted in order to gain a semantic equivalence. The hyponym of a predicate is more specific than the predicate itself and can be also useful. However, hypernyms are not so helpful, as one's interest in a certain concept does not necessarily imply his/her interest in a more general concept.

## 4 Future Research Directions

The above example introduced an initial work geared towards introducing an integrated architecture for *Situation Modeling* and *Smart Context Retrieval*, taking advantage of GUMO and UserML. A model for situated interaction and context-awareness was suggested, using WordNet and GUMO for supporting semantic conflict resolution of ubiquitous user and context models.

There are several kinds of conflicts that arise in a standard retrieval approach. Semantic conflicts are statements that are difficult to detect. If a system querying a ubiquitous user model is unaware of these problems it is hard to give correct recommendations based on the results of retrieval. The approach suggested above is based on semantic web technology and a complex conflict resolution and query concept, in order to be flexible enough to support adaptation in human-computer interaction in ubiquitous computing.

It is reasonable to presume that user modeling mediation may be enhanced by additional knowledge. The simplest example is the use of linguistic knowledge for identification of synonyms, antonyms etc. A step further may be the use of domain taxonomy for the mediation of user models. Furthermore, similarity of domains and mediation of domain-related user models from one domain in another domain requires additional knowledge about how to translate and interpret the information from the source domain in the target domain. This brings up the issue of what a domain is, how domains can be characterized and modeled, and how these definitions can be used for user modeling. The issue of context is another issue of great importance, since the same user may have different preferences for the same item in a different context [Berkovsky et al., 2006]. Figure 6 illustrates a semantically enhanced user modeling mediation, where, in addition to the specific mechanism used to transform the user modeling data from remote systems to the target system (may be regarded as a part of the catalogue suggested by Vassileva et al. [2003]), all various types of knowledge are used to select the right user attributes that are relevant to a given situation.

In the following list we sample and briefly discuss future research directions for ubiquitous user modeling that may help, in turn, to bridge the knowledge gap and allow building true ubiquitous user models, which may be stored on a user device or distributed in the environments:

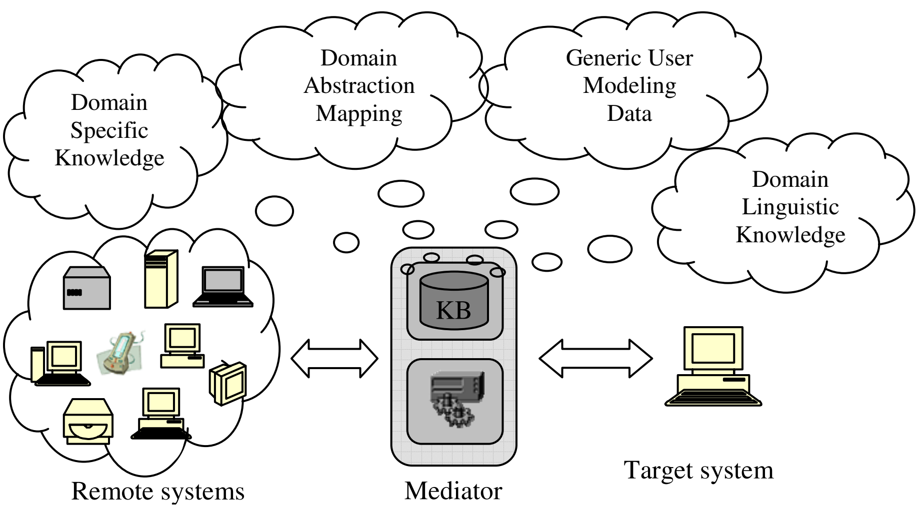


**Fig. 6.** Semantically enhanced user models mediation

- **Use of Domain Thesauri and Ontologies.** Different domains may have their own thesauri and ontologies. These are needed not only for user modeling purposes, but in general for standardization and common understanding of the domain. A good example is the medicine, where there is a systematic development of vocabularies. Although thesauri provide only linguistic information, they can standardize the domain terminology. Even WordNet can be used for identifying synonyms and enhance reasoning and mediation of content-based user modeling data. In particular, these ontologies can be used for a two-stage mediation: (1) bottom-up inference from the available user modeling data to the values of the ontology slots, and (2) inverse top-down inference from the inferred values of the ontology slots to the user modeling data required by the target personalization system.
- **Mapping GUMO Attributes to Specific Domains.** While GUMO provides a comprehensive user modeling ontology, different domains, situations, and even constraints of a certain situation may require to use different components of GUMO. Moreover, they may lead to different interpretations of the same slots of GUMO and their values. Hence there is a need to develop flexible mechanisms that will allow applying GUMO in different domains, situations, and constrains. This can be done, for instance, by a rule-based inference mechanism using GUMO attributes in specific conditions. These rules will lead to dynamically created localized views of GUMO, which can be applied for specific domains, situations, and constraints.
- **Contextual Aspects.** As already mentioned, different context may lead to different uses and interpretations of user modeling data. This is especially important for ubiquitous computing where the users' context changes frequently and dynamically. GUMO is naturally extensible for modeling various dimensions of contexts. This modeling, in turn, facilitates cross-context mediation of user modeling data, as suggested by Berkovsky et al [2006]. There, two complementary mediation types are presented: rule-based inference according to the rules crafted by domain experts or similarity-based reasoning applying statistical learning methods using previously collected user experiences.
- **Applying Machine Learning Approaches for Mediation Techniques.** Taking a closer look at content-based user modeling, a variety of machine learning techniques may be applied for user modeling purposes. Learning techniques used in the implemented mediation scenarios were quite simplistic and used intuitive reasoning mechanisms and shallow knowledge bases. However, this may hamper the accuracy of the derived user modeling data and, in turn, of the personalized services provided to the user. A natural question may be how to apply more accurate approaches and elicit the information using, for instance, Artificial Neural Network or the Support Vector Machine. While initial ideas were suggested by [Berkovsky et al., 2007], there are still a number of practical issues, machine learning approaches, and mediation scenarios to deal with.
- **Privacy Aspects.** With the evolvement of ubiquitous computing and user modeling, comes the issue of privacy. Personalized service requires the service provider to have a decent amount of personal information about the user, which can be provided by the user or by other systems, if the user is identified and allows such information transfer. Hence, mechanisms for preserving the privacy of the user and his/her personal information should be developed in parallel. However, the goals of

the privacy-preserving mechanisms contradict the goals of the personalization systems, leading to privacy versus accuracy trade-off. One possible compromise may be that the user will have a comprehensive representation of his/her model and allow parts of it to be provided anonymously to the service provider, if requested.

General user modeling ontology and user modeling mediation seem to be two orthogonal approaches to materialize the user modeling data interoperability in personalization systems. Each approach bears its own inherent advantages and limitations. This work presented a list of research issues that may help bridging the gap between the two approaches. We believe that incremental research efforts in these areas may gradually bridge the gap and allow applying both the semantics provided by GUMO and the user modeling mediation ideas in practice.

## Acknowledgements

This work has partly been supported by the EU-funded FP7 Project "GRAPPLE" which stands for "Generic Responsive Adaptive Personalized Learning Environment" and by the Australian Government through the Intelligent Island Program (administered by the Tasmanian Department of Economic Development and Tourism) and CSIRO. A special gratitude goes to Christian Blass and his work on semantic conflict resolution with WordNet in his Bachelor thesis [Blass, 2007].

## References

Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating Contextual Information in Recommender Systems using a Multidimensional Approach. ACM Transactions on Information Systems 23(1), 103–145 (2005)

Berkovsky, S., Aroyo, L., Heckmann, D., Houben, G.J., Kröner, A., Kuflik, T., Ricci, F.: Predicting User Experiences through Cross-Context Reasoning. In: Proceedings of ABIS 2006 - 14th Workshop on Adaptivity and User Modeling in Interactive Systems, Hildesheim, Germany, October 9-11 (2006)

Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of user models for enhanced personalization in recommender systems. User Modeling and User-Adapted Interaction 18(3), 245–286 (2008)

Blass, C.: Ubiquitous User Modeling: Semantic Conflict Resolution with WordNet, Bachelor Thesis, Saarland University (2007)

Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)

Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, U.S.A (1999)

Fellbaum, C. (ed.): WordNet – An Electronic Lexical Database. The MIT Press, Cambridge (1998)

Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition 5(2), 199–220 (1993)

Heckmann, D.: Ubiqutous User Modeling. Akademische Verlagsgesellschaft Aka GmbH, Berlin (2006) ISBN 3-89838-297-4 and ISBN 1-58603-608-4

Heckmann, D., Blass, C.: Context Integration for Ubiquitous User Modeling: Solving Semantic Conflicts with WordNet and GUMO. In: 5th International Workshop on Ubiquitous User Modeling, UbiqUM 2008, Gran Canaria, Spain (2008)

Heckmann, D., Krueger, A.: A User Modeling Markup Language (UserML) for Ubiquitous Computing. In: Brusilovsky, P., Corbett, A.T., de Rosis, F. (eds.) UM 2003. LNCS (LNAI), vol. 2702, pp. 393–397. Springer, Heidelberg (2003)

Heckmann, D.: Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource-adaptive computing. In: Hoto, A., Stumme, G. (eds.) LLWA Lehren - Lernen - Wissen - Adaptivität (ABIS 2003), Karlsruhe, Germany, pp. 283–286 (2003)

Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA (1999)

Jameson, A.: Modeling both the context and the user. Personal Technologies 5(1), 29–33 (2001)

Kay, J.: The UM toolkit for reusable, long term user models. User Modeling and User-Adapted Interaction 4(3), 149–196 (1995)

Kobsa, A.: Generic user modeling systems. User Modelling and User-Adapted Interaction Journal 11(1-2), 49–63 (2001)

Kray, C.: Situated Interaction on Spatial Topics. DISKI, vol. 274. Aka Verlag, Berlin (2003)

McCarthy, J., Buvac, S.: Formalizing context (expanded notes). In: Aliseda, A., van Glabbek, R.J., Westerstahl, D. (eds.) Computing Natural Language. CSLI Lecture Notes. Center for the Study of Language and Information, vol. 81, Standford University, CA, pp. 13–50 (1998)

McCalla, G., Vassileva, G., Greer, J., Bull, S.: Active Learner Modelling. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 53–62. Springer, Heidelberg (2000)

Mehta, B., Niederee, C., Stewart, A., Degemmis, M., Lops, P., Semeraro, G.: Ontologically-Enriched Unified User Modeling for Cross-System Personalization. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 119–123. Springer, Heidelberg (2005)

Mehta, B., Nejdl, W.: Intelligent Distributed User Modelling: from Semantics to Learning. In: UbiDeUM, Corfu, Greece (2007)

Morita, M., Shinoda, Y.: Information Filtering Based on User Behavior Analysis and Best Match Retrieval. In: Proceedings of the Annual International Conference on Research and De-velopment in Information Retrieval, Dublin, Ireland (1994)

Muhammad, T., Vassileva, J.: Policies for Distributed User Modeling in Online Communities. In: Proc. Workshop UbiDeUM 2007, at the 11th International Conference UM 2007, Corfu (2007)

Pease, A., Niles, I., Li, J.: The suggested upper merged ontology: A large ontology for the semanticweb and its applications. In: AAAI 2002 Workshop on Ontologies and the Semantic Web. Working Notes (2002), `http://projects.teknowledge.com/AAAI-2002/Pease.ps`

Stahl, C., Heckmann, D.: Using semantic web technology for ubiquitous location and situation modeling. The Journal of Geographic Information Sciences CPGIS: Berkeley 10, 157–165 (2004)

van Assem, M., Gangemi, A., Schreiber, G.: Rdf/owl representation of wordnet. Technical report, Vrije Universiteit Amsterdam (2002)

van der Sluijs, K., Houben, G.J.: Towards a Generic User Model Component. In: Workshop on Personalization on the Semantic Web (PerSWeb 2005), at the 10th International Conference on User Modeling (UM 2005), Edinburgh, Scotland, July 25-26, pp. 43–52 (2005)

Vassileva, J., Greer, J.E., McCalla, G.I.: Openness and Disclosure in Multi-agent Learner Models. In: Proceedings of the Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling, International Conference on AI in Education, Lemans, France (1999)

Vassileva, J.: Distributed User Modeling for Universal Information Access. In: Stephanidis, C. (ed.) Universal Access in Human - Computer Interaction (UAHCI), Proceedings of the 9th International Conference on Human-Computer Interaction, New Orleans, USA, vol. 3, pp. 122–126. Lawrence Erlbaum, Mahwah (2001)

Vassileva, J., McCalla, G., Greer, J.: Multi-Agent Multi-User Modeling. User Modelling and User Adapted Interaction 13(1), 1–31 (2003)

Yimam-Seid, D., Kobsa, A.: Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. Journal of Organizational Computing and Electronic Commerce 13(1), 1–24 (2003)