User Model Interoperabily: A survey

Published version:

DOI:10.1007/s11257-011-9097-5

Noname manuscript No.
(will be inserted by the editor)

User Model Interoperability: a Survey

Francesca Carmagnola, Federica Cena and Cristina Gena

Received: date / Accepted: date

Abstract Nowadays a large number of user-adaptive systems has been developed. Commonly, the effort to build user models is repeated across applications and domains, due to the lack of interoperability and synchronization among user-adaptive systems. There is a strong need for the next generation of user models to be interoperable, i.e. to be able to exchange user model portions and to use the information that has been exchanged to enrich the user experience.
This paper presents an overview of the well-established literature dealing with user model interoperability, discussing the most representative work which has provided valuable solutions to face interoperability issues. Based on a detailed decomposition and a deep analysis of the selected work, we have isolated a set of dimensions characterizing the user model interoperability process along which the work has been classified. Starting from this analysis, the paper presents some open issues and possible future deployments in the area.

Keywords user model interoperability, user modeling, interoperability, user-adaptive systems

1 Introduction

Nowadays user-adaptive systems [Brusilovsky et al., 2007] are used in different areas, from e-commerce to e-learning, from tourism and cultural heritage to digital libraries, etc. A user-adaptive system adapts its contents, structure and interface according to the user features contained in the user model. The user model typically maintains user properties such as preferences, interests, behavior, knowledge, goals and other facts that are deemed relevant for a user-adaptive application [Brusilovsky, 1996, Kobsa et al., 2001]. A user model is often conceived as an overlay of the domain model, where the user's current state with respect to domain concepts is recorded [Brusilovsky, 1996, 2001]. Adaptive applications usually maintain a specific user modeling component, which is in charge of incrementally constructing a user model by storing, updating and deleting entities and supplying the other

F. Carmagnola, F. Cena, C. Gena
Dipartimento di Informatica, Università di Torino
Corso Svizzera 185; 10149 Torino, Italy
E-mail: carmagnola, cena, cgena@di.unito.it

components with assumptions about a user [Wahlster and Kobsa, 1989]. The user model is a key component of an adaptive system. Indeed, the quality of personalized services provided to the user largely depends on the characteristics of the user model, like its accuracy, the amount of data it stores, whether such data are up to date, etc.

The great proliferation, both of user-adaptive systems and of systems that gather personal information about the user, such as social web sites (e.g. Del.icio.us[1], Flickr[2], YouTube-Flickr[3], etc.) and mobile applications (e.g., the iPhone and iPod family) leads to the replication of user data, such as preferences, knowledge, activities, tags, etc, over many applications. These considerations lead to a great challenge: development of environments where user-adaptive systems cooperate. We refer to this process as *user model (UM) interoperability*.

Achieving interoperability in an open and dynamic environment like the Web is a difficult and complex task and requires a very high level of alignment by applications [Aroyo et al., 2006]. For this reason, only a limited number of adaptive applications really cooperate to share UM knowledge, for example InterBook and PAT Online [Brusilovsky et al., 1997], i-Help and S-UM [McCalla et al., 2000], MOT and WHURLE [Stewart et al., 2006], Trip@dvice and PIL [Berkovsky et al., 2006a], CHIP and iCITY [Wang et al., 2008].

The paper analyses UM interoperability focusing on the challenges and requirements needed for the user modeling process in the context of distributed applications on the Web. We present an overview of the most relevant work dealing with UM interoperability which have been proposed in recent years conferences and journals. Furthermore, we have identified a set of dimensions along which individual systems can be classified. We hope that such dimensions help researchers approaching the field and could be used as a guide for further work in UM interoperability.

The paper is structured as follows. We first discuss what UM interoperability means, what it implies, and how it can be performed. Then, we present the main motivations for systems interoperability on the Web, and for UM interoperability in particular (Section 2). Section 3 presents a brief history of the move from centralized to decentralized user models, focusing on the architectures for UM interoperability, in terms of conceptualisation and storage of the user model. In Section 4 we present the review methodology we have followed. We have first selected and described a set of work by screening the relevant literature on UM interoperability. Then, based on a detailed decomposition of the selected work, we have isolated a set of relevant dimensions. A separate section is devoted to each dimension: *languages and protocols for communication* (Section 5), *the kind of data that can be exchanged* (Section 6), *how the exchanged data can be represented* (Section 7), *possible usage and integration of the exchanged data* (Section 8), and *privacy issues* (Section 9). For each dimension, we present the approaches and techniques exploited by the work we have analyzed. Section 10 briefly presents some aspects which have not been covered in this survey. Finally, the last section (Section 11) concludes the paper by discussing some challenging open issues and future trends for UM interoperability with the aim of inspiring promising research directions.

[1] http:/el.icio.us/

[2] http://www.flickr.com/

[3] http://www.youtube.com/

2 Interoperability: definition and motivations

As stated in the introduction, we address the issue of *UM interoperability* as the process of exchanging distributed user data across applications. A formal definition of *interoperability* was provided by Wegner who defined interoperability as "the ability to cooperate and exchange data despite differences in languages, interface and execution platform" [Wegner, 1996]. Overcoming such differences is a challenging task which deals with three main aspects: structure, language and logic. *Structural interoperability* concerns the possibility to bridge the differences between information systems at the access level. *Syntactic interoperability* (also referred to as *language interoperability*) refers to the capability of different systems to interpret the syntax of the data in the same way. *Semantic interoperability* (also called *logical interoperability*) is the ability of systems to exchange the information on the basis of shared, pre-established and negotiated meanings of terms and expressions. Thus, it concerns the possibility to bridge differences between information systems on the meaning level.

Structural interoperability can be accomplished by exploiting shared communication protocols and standardized interfaces for accessing the data. Enabling syntactic and semantic interoperability is challenging since it requires a high degree of alignment among the applications. As a matter of fact, different systems may represent the same data in different ways, using different syntactic and conceptual structures, using different terminologies or different interpretations of the same terminology [Staab and Stuckenschmidt, 2006]. Therefore, some kind of agreement and clarification of the data among the systems is required.

Data interoperability has been considered very useful in different contexts. The experience in industry showed that interoperable systems lead, to some financial benefits[4], such as:

- *lower costs per transaction*. Information sharing when standardization of interfaces is available may facilitate application integration and data exchange;
- *increased operating efficiency.* The interoperability of systems often brings a reductions of the number of devices since they can be shared among many different systems. Thus, the overall cost of the system decreases;
- *higher quality service levels and more predictable response.* The ability of multiple parties to share information on the status of constituents and systems is facilitated by interoperability;
- *data creation and information integration.* The definition of new data or the integration of previously scattered pieces of information is enabled by interoperability;
- *increased competition for customers.* When different products can be easily combined without expensive interfaces, companies can make specialized products. In this sense, interoperability supports innovation.

Besides the financial advantages, in general data interoperability allows i) acquiring missing data and more accurate data, and (ii) achieving functionalities that systems do not implement by themselves. The first point i) deals with the ability of the interoperable systems or components to exchange information and to use the information that has been exchanged. The second point ii) deals with the capability of products, systems or business processes to work together to accomplish a common task. This represents the possibility of a system module to be replaced by another module belonging to a different application, allowing applications to delegate functionalities.

[4] http://www.gridwiseac.org/pdfs/financial_interoperability.pdf

The above discussed general motivations for interoperability can be revisited for UM interoperability as i) acquiring more data and more accurate data about users, ii) acquiring functionalities (both user model and user modeling functionalities) that systems do not themselves implement. Many authors have discussed the main motivations for UM interoperability, focusing on those resulting from exchanging the user model data.

Kobsa [2007b] envisions in UM interoperability a way to speed up the phase of the user model initialization, when the user model does not store enough data to provide appropriate adaptation. This is the situation of the so-called "cold start problem" which refers to the difficulty for applications to start up the adaptation for new users. For example, in CHIP and iCITY [Wang et al., 2008], systems communicate and exchange user data to solve the cold-start problem. The cold start problem is particularly relevant when the user interacts with adaptive educational hypermedia systems. Authoring, as well as maintaining of adaptive educational hypermedia systems, can be an extremely complex and a time consuming task. Considering these difficulties, the idea of providing interoperability between different systems turns out to be not only desirable but also necessary, as this will enable the re-use of previously created materials without the cost of recreating them from scratch. Examples of interoperable adaptive educational hypermedia systems are MOT and WHURLE [Stewart et al., 2006], i-Help and S-UM [McCalla et al., 2000], InterBook and PAT Online [Brusilovsky et al., 1997].

Vassileva [2001] states that UM interoperability relieves users from the pain of training new systems. Users typically do not appreciate wasting time filling in their model in every application they use. However, every time a user interacts with a system for the first time, she needs to provide the data that could have already been supplied to other applications.

Berkovsky et al. [2005] assume that UM interoperability enriches the user model knowledge stored in a system (*qualitative* improvement). In [Wang et al., 2008], CHIP and iCITY systems communicate and exchange user data also to get a more exact view of the users interests.

Heckmann [2005b] makes reference to an increased amount of information about the users due to the opportunity to benefit from the efforts led by other modelers and systems. This leads to the "increased coverage" of the model, since more aspects can be covered by the aggregated user model, including in the user model the features that one system could not acquire by itself (*quantitative* improvement). This qualitative and quantitative improvement of the user model data has the potential to give better adaptation results. There are examples of systems which interoperate to reach a better level of adaptivity, such as InterBook and PAT Online [Brusilovsky et al., 1997], and Trip@dvice and PIL [Berkovsky et al., 2006a].

As a final remark, based on the above discussed motivations, we can distinguish different kinds of systems with different features and architectures which may benefit from UM interoperability for different purposes. First, for systems that collect and use user information without being user-adaptive (thus without User Model and User Modeling Components), we envision in UM interoperability a way to provide adaptation even in absence of user model data. Second, for systems which have User Model Component storing user data but do not have a User Modeling Component, we foresee in UM interoperability an opportunity for making use of an external user model architecture and reasoning over the data. Finally, for systems which have both a User Model Component and a User Modeling Component, we see in UM interoperability a way to obtain more data from a qualitative and quantitative perspective. Moreover, regardless of the presence of the User Model or User Modeling Component, UM interoperability enables those applications working on mobile devices, thus with limited memory and computing power, to make use of user model data stored in external systems.

3 From centralized to decentralized UM interoperability

In this section, we present and discuss the steps which have been taken in the user modeling community toward UM interoperability, with particular attention to the shift from centralized to decentralized user model architectures. We distinguish the *physical storage* of the user model, where the user data are physically maintained (the cylinders in Fig. 1, 2 and 3) from the *conceptualization of the model*, that is how the user model component is conceived in terms of being shared or not between systems (the rumbles in Fig. 1, 2 and 3). This allowed us to differentiate systems which are both physically and conceptually centralized (centralized approach), systems which are both physically and conceptually decentralized (decentralized approach), and systems which are physically decentralizedand conceptually both centralized and decentralized (mixed approach).

3.1 Centralized approaches

The first step toward UM interoperability was made by *generic user modeling systems* (also known as *user modeling shell systems*). According to Kobsa, a user modeling system is defined as "generic user modeling system" if it is independent from the architecture and from the user model of a specific user-adaptive application [Kobsa et al., 2001]. Generic user modeling systems serve as a separate user modeling component in an application system at runtime, and developers are simply required to fill the shell systems with the application-specific user modeling knowledge. Examples of user model shell systems are UMT [Brajnik and Tasso, 1994], TAGUS [Paiva and Self, 1995], um [Kay, 1995], BGP-MS [Kobsa and Pohl, 1995]. Generic user modeling systems had the aim of allowing the reuse of user modeling components which have been already developed. Thus, they have focused on the technological reuse rather than on the reuse of the user model data collected by the systems.

However, developing a user-adaptive system is a challenging task requiring much effort not simply related to the development of user modeling components. A great effort is devoted to the management and integration of the system knowledge base (about users, domain, context, etc.). *User modeling servers* have been designed to reduce the effort in knowledge management, offering a flexible client-server architecture. As shown in Fig. 1, in a user modeling server setting, a user model (conceptually and physically conceived as a *centralized repository*) is maintained by and shared across several applications through a flexible client-server architecture. User modeling servers work as an application-external knowledge base. In this way, the knowledge about the user is made available for more than one application at the same time and user information acquired by one application can be employed by other applications (see Fig. 1 and [Fink, 2003] for more details). Both user modeling servers and shell systems separate the components for user modeling from other components. However, many features differentiate them. According to Fink and Kobsa [2000], the client-server architecture of user modeling servers is the most distinctive feature. Furthermore, user modeling servers are not functionally integrated into an application, but they usually belong to a local or wide area network and can serve more than one application at the same time. Furthermore, user modeling servers present advanced features with respect to shell systems: offering an architecture to manage, store and query the user model efficiently and let it be accessible by many systems at a time, and allowing systems to benefit from collecting more user data compared against the user data gained by an isolated system.
Several user modeling servers have been proposed in recent years in several different domains. Examples of foremost user modeling servers are: DOPPELÄNGER [Orwant, 1995];

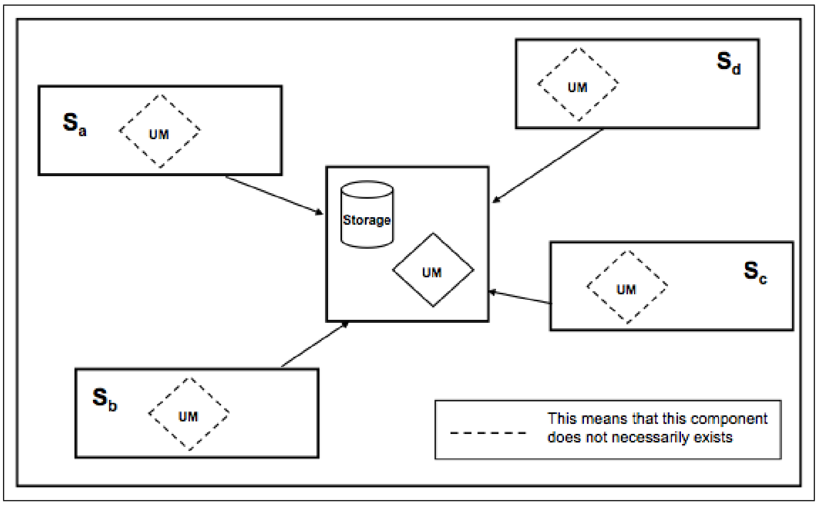

Fig. 1 Centralized approach to UM interoperability.

Learn Sesame [Caglayan et al., 1997]; GroupLens [Konstan et al., 1997]; LMS [Machado et al., 1999]; PersonisAD [Assad et al., 2007]; MEDEA [Trella et al., 2003]; Cumulate [Brusilovsky, 2004]; UMS [Kobsa and Fink, 2006][5].

Despite the evident benefits of centralized user modeling systems, they show some potential weaknesses [Kobsa, 2007b]. First of all, user modeling systems impose a centralized user model, which can be sometimes very restrictive. Secondary, they have well-defined points of access, at which information about the user and requests for user model entries can be submitted and where answers of the user modeling server can be received. Consequently, there is a central point of failure. While reliability can be increased by introducing mirrors or distributing the information on several servers[6], the problem of synchronization and coordination of the mirror servers increases the cost. Another problem is related to the protection of the user data stored at a unique point.

3.2 Decentralized approaches

The above seen limitations led to the definition of a new approach for managing UM interoperability, the *decentralized approach* [Vassileva, 2001, Heckmann, 2005]. While in the centralized approach, a single user model is shared and enriched by the interacting systems, in decentralized settings each system maintains a small user model, as needed for its own purposes of adaptation. Thus, there is a collection of user model fragments distributed among the systems the user interacts with. This is even more valid in ubiquitous computing and intelligent environments [Weiser, 1998], where there is the need for new kinds of

[5] For a more detailed discussion about the characteristics and limitations of user modeling servers, the interested reader can refer to [Kobsa, 2001, Fink, 2003, Kobsa, 2007b].

[6] This is known as *virtual centralization* of distributed user models, where there is a unique user model but different parts of it are stored separately physically on different servers. Fink and Kobsa ([2006]) pointed out that centralized user modeling does not necessarily imply the physical centralization of user information. The user information is replicated and thus this improves the performance [Kobsa and Fink, 2006].

user modeling architectures since numerous unrelated sensors and devices acquire limited information about users [Lorenz, 2005]. Decentralized user modeling investigates how to combine partial user data and make sense of it in a specific context [Dolog and Vassileva, 2005]. As shown in Fig. 2, in decentralized user modeling applications have their own physical and conceptual representation of the user model (respectively the cylinder and the rumble in every system). However, they communicate directly in order to exchange user and domain data in a peer-to-peer manner. Thus the storage, as well as the conceptualization of the user model, are purely decentralized (see Fig. 2).

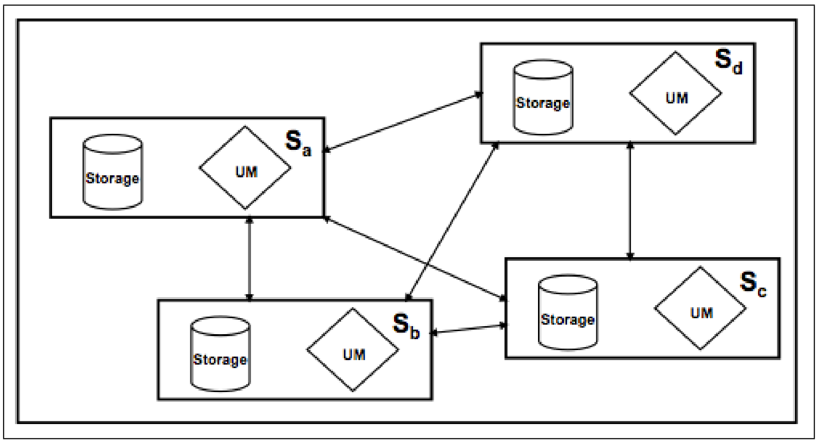

Fig. 2 Decentralized approach to UM interoperability.

The issue of decentralized user modeling has been variously addressed in recent years, due to the fact that the model of communication in intelligent environments calls for decentralization [Fink, 2003, Heckmann et al., 2005b]. The solutions proposed to UM interoperability in a decentralized setting offer more functionalities compared with centralized solutions. For instance, each individual system may define its own privacy policy about the part of user model to be shared, keeping private some portion of the user model and defining different access rules according to requestor reputations [Kobsa and Schreck, 2003] (see Section 9). Usually decentralized user modeling solutions also offer functionalities for the mapping and the integration of different knowledge models, since owners have the possibility to represent and organize knowledge in the most appropriate way and the knowledge exchange can be carried out without assuming shared meanings but rather enabling the dynamic translation of different meanings. They often provide solutions for the discovery of communication partners and services.
Examples of decentralized approaches are: Zang et al. [2006], Lorenz [2005], Metha et al. [2005], Heckmann [2005], Carmagnola and Dimitrova [2008], Niu et al. [2003], Dolog and Schafer [2005a], Brooks et al.[2004], and Cena and Furnari [2009].

3.3 Mixed approaches

There is a possibility for systems to implement **mixed solutions**, where the user models are physically decentralized, and conceptually both centralized and decentralized. More specifically, in such solutions, the systems store their user model locally, being decentralized in this sense (see in Fig. 3 the cylinders and the rumbles in every system). However, UM interoperability is ensured by referring to a centralized model which includes the most used concepts within the domain. Thus, from a conceptual point of view, such systems are centralized. An example of such an approach is Berkovsky [Berkovsky, 2006], where the models are stored in a decentralized way by the service providers, but mediation is done through a centralized point of access. Other examples are GUC [van der Sluijs and Houben, 2006] and Medea [Musa and de Oliveira, 2005]. Notice that Dolog and Schafer [Dolog and Schäfer, 2005a] provide an API which allows for a mixed approach or a purely centralized solution. It is up to the application programmer to decide.

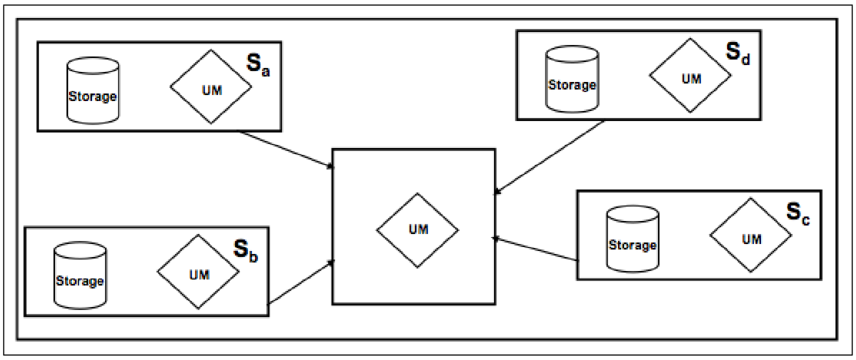

Fig. 3 Mixed approach to UM interoperability.

To summarize, in the *centralized approach* all the user information is stored in a central repository (*centralized storage* or virtually centralized, where the data are replicated among several servers). Also the model is unique, and thus the conceptualization of the models is centralized *(centralized model)*.
In the *decentralized approach*, each system is in charge of managing its local repository storing the user model and it communicates with the other systems to collect user model data when needed. In this case, both the storage and the conceptualization of the models are decentralized *(decentralized model/decentralized storage*).
Finally, in the *mixed approach*, there are scattered local storages of user data (*decentralized storage*) and some different local user models *(decentralized model)* which refer to a shared model in a conceptually centralized perspective *(centralized model*).

4 Review methodology

As discussed in Sections 2, UM interoperability is a challenging task dealing with several issues. This review aimed at presenting and discussing the most representative work that has provided valuable solutions to these issues. To this end, we screened a vast number of work in the literature. Most of the examples reported in the paper refer to the work presented in International Conferences dealing with User Modeling, with particular attention to the User Modeling and Adaptive Hypermedia conferences from 1993 and 2010. In addition, we included work either with a special focus on interoperability or prior interoperability

surveys published by international journals, such as the journal of User Modeling and User-adaptive Interaction[7], from 1997 to 2009. Note that we have selected the work according to the bottom-up approach: we started from the main previous work on interoperability (collected and analyzed in the Ph.D. theses of the two authors Carmagnola [2007], Cena [2007]) and then we have moved towards the work cited in them, selecting all the relevant issues and examples.
Considering such a huge literature, we have isolated a set of representative work including existing systems, frameworks and approaches, and classified them on the basis of their main interoperability tasks. Three main categories have emerged: systems whose principal goal is *to exchange user/learner models*; systems whose principal goal is to provide a *user modeling/adaptation service*; systems whose principal goal is *to share user models across applications*.

In the following we provide a brief description of all the work we have analyzed, classifying it according to its main interoperability traits.

Systems that exchange user models aims to facilitate the exchange of user models, or part of them, across different distributed applications.The exchange takes place on demand. This can be useful when applications have incomplete or partial data about the user.

- **Niu et al.** [2003]: an approach to decentralized user modeling where the exchange of user models is on demand. In this approach at run-time several context-dependent user modeling processes are exploited. The user model is created on the fly, thus fixed storage of the user model is not required. The Gnutella P2P protocol is used for discovering user model data residing in different systems/agents;
- **Lorenz et al.** [2005]: a framework for distributed user modeling in ubiquitous computing where a team of agent systems cooperates to exchange the information about the user and her context through brokering the components hosted on the distributed local devices. To enable *communication* and *service discovery* between the agents the authors foresee a mix of the blackboard and the message-sending approaches;
- **Dolog and Schafer** [2005a]: the conceptualization and implementation of a framework which provides a common base for the exchange of learner profiles between several sources. The exchange representation of learner profiles is based on standards for profiling learners (such as IEEE PAPI [PAPI, 2003]) and IMS LIP [LIP, 2005]). Different public interfaces (*APIs*) are designed and implemented to create/export and manipulate such learner profiles;
- **Musa and Palazzo Moreira de Oliveira** [2005]: a web service based architecture for the exchange of the learner's model information between e-learning systems on the web. As a result, the course content, in any federated e-learning system is adapted and presented to students, according to each student's program, cognitive characteristics, and navigation preferences. The exchange is based on standards for profiling learners (such as PAPI and LIP);
- **Mehta et al.** [2005]: an approach for cross-system personalization relying on a unified profile that contains several aspects of a user and it is stored inside a "Context Passport" that the user carries along in her journey across information space. The Context Passport can be exploited also for *user identification*. As a basis for the exchange of user profile information between multiple systems, the authors also define an ontology-based user context model, the *Unified User Context Model* (UUCM). The authors also provide a

[7] http://www.umuai.org/

privacy preserving distributed framework to protect user data by means of encryption of data;
- **Heckmann** [2005a]: an architecture for decentralized user modeling where user-adaptive systems can exchange user information by means of decentralized user model services. This user model service can be distributed within the intelligent environment. The authors also propose the *General User Model Ontology* (GUMO) for the uniform interpretation of decentralized user models as well as the RDF-based UserML language for the communication between applications to exchange user models. Users also have control on privacy management via a user model editor;
- **Zhang et al.** [2006]: a web service-based architecture and an ontology-based user model for cross-system personalization. The authors also define the User Role Model ontology to foster the reuse of user models among systems;
- **Van der Sluijs and Houben** [2006]: GUC, the Generic User Model, a generic component for exchanging user models between web-based systems. GUC provides the application with storage facilities for the user models, and it also allows the configuration of a distributed management of *semantic mappings* between user models. To reach these goals, GUC uses semantic web and web service technologies. The framework also allows users to manage privacy preferences;
- **Carmagnola and Dimitrova** [2008]: a framework for UM interoperability which deals with semantic heterogeneity of user models and automates the user model exchange across applications. The framework provides *semantic mapping* of the user data from one system onto another one. For this purpose, the authors implement an algorithm based on evidential reasoning and advances in the semantic web;
- **Berkovsky et al.** [2008]: a general framework and specific methodologies for enhancing the accuracy of user modeling in recommender systems by importing and integrating the data collected by other recommender systems. Such a process is defined as *user models mediation* and it is performed by a mediation component. The component provides support by identifying semantic relations between different concepts in the domain. The framework also provides a privacy preserving solution by means of perturbation of data;
- **Cena and Furnari** [2009]: a model for achieving UM interoperability by means of semantic-based dialogues.The exchange of information between systems takes place in a P2P manner, by means of atomic communication (if systems share the same knowledge model) or by means of negotiation techniques based on dialogue models, for the *mapping* between different knowledge models. The framework also provides a centralized registry for *systems discovery*. This conceptual model has been implemented in a framework built on a SOA-based environment.

Systems that provide a user modeling/adaptation service are central servers aimed at i) receiving a defined set of data from others (applications/systems), ii) performing some kind of elaboration on the received data, iii) providing services of user modeling and/or adaptations/recommendations usually sending a defined set of data. These services - supplied in real time - are extremely useful when the systems requiring the user model do not implement user modeling/adaptation functionalities. These systems do not share user model information between different applications; thus, the supplied service is a one-to-one service.

- **GroupLens** [Konstan et al., 1997]: a collaborative filtering system that provides users with Usenet news they are predicted to be interested in. GroupLens calculates the interests of the users in some articles, starting from the interests of similar users. The clients can connect to the Usenet server to retrieve articles and to the GroupLens server to obtain prediction of users' interests for those articles;

- **Learn Sesame** [Caglayan et al., 1997]: a user modeling server derived from the former Open Sesame learning agent. The server is able to perform the following tasks: i) it collects implicit and explicit user, usage, and environmental data, ii) it analyzes the stream of events by means of clustering techniques, iii) if it recognizes some regularities, it supplies client applications with evidence for such regularities;
- **PersonisAD** [Assad et al., 2007] is an evolution of Personis [Kay et al., 2002], from which it inherits many features and functionalities. PersonisAD is is a framework for building context-aware, ubiquitous applications, which supports distributed models with active elements which can be triggered when relevant events occur. It provides a powerful and consistent means to respond to significant changes in the models of people, sensors, devices and places. Finally, it also supports distributed models and associated resource discovery;
- **MUMS** (**Massive User Modelling System**) [Brooks et al., 2004]: a framework and prototype for supporting the just-in-time *production, delivery and storage* of user modeling information. The authors use MUMS in e-learning environments, but the system itself is agnostic with respect to the information being shared. Any domain that can be expressed in RDF/OWL can be used. The specific implementation for the MUMS general architecture is web services-based and exploits semantic web techniques and languages;
- **UMS** (**User Modelling Server**) [Kobsa and Fink, 2006]: a user modeling server based on the Lightweight Directory Access Protocol (LDAP). UMS allows for the representation of different models (such as user and usage profiles, and system and service models), and for the inclusion of arbitrary components that perform user modeling tasks on these models. External clients, such as user-adaptive applications, can submit and retrieve information about users. LDAP also provides a support for user *privacy* protection and *user identification*.

Systems that share user models across applications are central servers that can be accessed by several applications that are provided with the requested user/learner model or a part of it, in a one-to-many exchange of information. These systems are aimed at i) collecting data about the user in different ways and/or from different sources, ii) elaborating the data, iii) integrating such data to form a richer model of the user, and iv) making the resulting information available to ”any” applications requiring it (notice that they are proactive in gathering and elaborating user data, and this is what distinguishes them from systems providing user modeling/adaptation services). Sharing a user model across applications can be also useful when applications do not implement user modeling/adaptation functionalities.

- **DOPPELÄNGER** [Orwant, 1995]: a generalized user modeling server that i) uses hardware and software sensors to gather data about the users; ii) makes inferences about the data through heterogeneous learning techniques, and iii) makes the resulting information available both to applications and to users. Clients systems can communicate with DOPPELÄNGER to retrieve information about the users. Finally, users can control the access to any of all parts of their user models;
- **LMS** (**Learner Modelling Server**) [Machado et al., 1999]: a server that can be accessed by different applications in a multi-agents platform in the e-learning domain. It allows the reusability of the software components (user modeling process) and the sharing of the learning models. It can be used by several applications that can share parts of the learner models if they use the same shared domain ontology;
- **MEDEA** [Trella et al., 2003]: an open, service-based, learning platform for the development of intelligent web-based educational systems. It allows use of different types of resources for intelligent instruction purposes, even those that do not include the learner

	Centralized approach	Decentralized approach	Mixed approach
Exchange of user/learner models		Niu et al, 2003 Lorenz, 2005 Dolog and Schfer, 2005 Metha et al, 2005 Heckmann, 2005 Zhang et al, 2006 Carmagnola and Dimitrova, 2008 Cena and Furnari, 2009	Musa and Palazzo Moreira de Oliveira, 2005 Berkovsky et al, 2008 Van der Sluijs and Houben, 2006
User modeling/adaptation service	Group Lens, 1997 Learn Sesame, 1997 PersonisAD, 2007 MUMS, 2004 UMS, 2006		
Share user models across applications	DOPPELANGER, 1995 LMS, 1999 MEDEA, 2003 CUMULATE, 2004		

Table 1 Combination of interoperability tasks and architectures.

model of the client applications. The MEDEA platform is defined to allow the integration, at runtime, of any web-based tool that could be encapsulated as a web service;

- **CUMULATE** [Brusilovsky, 2004, Yudelson et al., 2007], a student modeling server that implements KnowledgeTree [Brusilovsky, 2004], an architecture for adaptive e-learning based on distributed reusable intelligent learning activities. CUMULATE i) collects evidence (events) about students from multiple servers that interact with the students; ii) processes these events; and iii) makes the results of the processing available to other systems by means of simple HTTP/GET request - XML reply.

Table 1 classifies the above described work by combining their main interoperability features with their architectures (Section 4). By analyzing it, it becomes apparent that there is a strict link among interoperability tasks and architectures. On the one hand, systems whose interoperability task is to facilitate the exchange of user/learner model across different distributed applications implement a decentralized or a mixed architecture, where systems can directly communicate. On the other hand, systems whose interoperability task is to provide a user modeling/adaptation services are characterized by the fact that they are able to receive a defined set of data from other systems, performe inferences and provide user modeling/adaptation services. They implement a centralized architecture where a centrally stored user model is maintained by different applications through a flexible client-server architecture. Finally, work sharing user models across applications implement a centralized architecture as well. However, they do not have a service-oriented behavior in term of request/response.

A detailed analysis of the above systems revealed a number of common features. For example, in all the work we have considered, specific solutions for enabling communication among distributed systems are discussed. Moreover, many have focused on facing the heterogeneity of distributed user and domain data. Some others have managed the protection of user's privacy during the interoperability process. We grouped these common features in *dimensions* characterizing the UM interoperability process. We hope that such characteristics could serve as a guide both for further studies in the field and for the design of interoperable user modeling systems. Indeed, based on the analyzed work, the dimensions we have isolated are:

- **Protocols for communication**: the languages and protocols employed for communicating with the other systems with the aim of exchanging user model data;
- **Exchanged data**: which kind of data are exchanged;
- **Representation of the exchanged data**: how the exchanged user models are represented from a syntactic and semantic perspective;
- **Integration of the exchanged data**: whether the user model data collected by the interoperability process are integrated into an existing knowledge structure or they are computed just in time as needed; also how systems deal with possible conflicts;
- **Privacy**: how to manage the privacy issue in the context of UM interoperability.

In the next sections each of these dimension will be discussed by presenting all the solutions which have been proposed in the work we have analyzed. For every dimension of the analysis, we present a table which classifies the work we have analyzed with respect to the dimensions. The work is grouped in the following tables according to the interoperability task they support. Moreover, we further classify the systems into sub-groups according to similar features. More specifically: Niu et al. [2003] and Lorenz [2005] both provide run-time creation of user models; Dolog et Schafer [2005] and Musa and Palazzo Moreira de Oliveira [2005] exploit standards for profiling learners (PAPI and LIPS); Mehta et al. [2005], Heckmann [2005] and Zhang et al. [2006] propose *ad hoc* ontologies for UM interoperability; Van der Sluijs and Houben [2006]; Carmagnola and Dimitrova [2008], Berkovsky et al. [2008] and Cena and Furnari [2009] provide solutions to support mapping among ontologies in UM interoperability context; finally, all the remaining systems provide user model and modeling service functionalities.
As a final remark we note that every section also contains, in addition to the set of above identified work, examples taken from other work that is relevant only for the specific dimension described in the section.

5 Communication

First of all, in order to exchange user model data, systems need suitable protocols to support interoperable machine-to-machine interaction (Section 5.1). Then, at higher level, they can exchange user data exploiting different languages (Section 5.2).

5.1 Protocols for communication

Adherence to *open standards protocols* for communication facilitates interoperability. Applications can use a standard for remote calls (i.e., remote procedure call or remote methods invocation) or web-based protocols.

Remote calls. UM data can be exchanged using the traditional standards for Remote Procedure Call[8].
Applications can provide some specific remote procedure calls to enable other systems to access their user model data. Examples are the different APIs provided by user modeling

[8] A remote procedure call (RPC) is an Inter-process communication that allows a computer program to cause a procedure to execute in another address space without the programmer explicitly coding the details for this remote interaction.

	Standard for communication	Languages
Exchange of user/learner models		
Niu et al, 2003		
Lorenz, 2005		UserML
Dolog et Shafer, 2005	Web Service	
Musa and Palazzo Moreira de Oliveira, 2005	Web Service	
Metha et al, 2005		CSCP
Heckmann, 2005	Web Service	UserML
Zhang et al, 2006	Web Service	
Van der Sluijs and Houben, 2006	Web Service	
Carmagnola and Dimitrova, 2008		
Berkovsky et al, 2008		
Cena and Furnari, 2009	Web Service	Dialogue Game
User modeling/adaptation service		
Group Lens, 1997	http	
Learn Sesame, 1997	http	
PersonisAD, 2007	http	JSON
MUMS, 2004	Web Service	
UMS, 2006	LDAP, CORBA	
Share user models across applications		
DOPPELANGER, 1995	TCP/IP	
LMS, 1999	Java RMI	KQLM
MEDEA, 2003	Web Service	
CUMULATE, 2004	HTTP get request/XML reply XML request/XML reply	

Table 2 Protocols and Languages for Communication.

servers to access their knowledge, both about users and domain[9].
LSM [Machado et al., 1999] is organized as a client-server model, and relies on Java remote service invocation (RMI) protocol[10].
In Learn Sesame [Caglayan et al., 1997] the communication between components is carried out via CORBA[11].
In UMS [Kobsa and Fink, 2006], the internal components (the directory component and the user modeling components) communicate via LDAP[12] and CORBA. Such components can be distributed across a network. The separation of event handling and information access at different layers allows one to change the storage modalities (e.g., replacing the LDAP-based information management with a SQL-based management), while still preserving the CORBA-based communication layer. The communication between external clients and the UMS Directory Component is carried out through LDAP and through ODBC for those clients that are not LDAP enabled.

[9] The interested reader can refer to Fink and Kobsa ([2000]) for a complete list of user modeling systems and their specific APIs.

[10] The Java Remote Method Invocation Application Programming Interface (API), or Java RMI, is a Java application programming interface that performs the object-oriented equivalent of remote procedure calls.

[11] The Common Object Requesting Broker Architecture (CORBA) is a standard defined by the Object Management Group (OMG) that enables software components written in multiple computer languages and running on multiple computers to work together. CORBA is a mechanism in software for normalizing the method-call semantics between application objects that reside either in the same address space (application) or remote address space (same host, or remote host on a network).

[12] LDAP, the Lightweight Directory Access Protocol [Howes et al., 1999], is a standardized protocol for accessing information about relevant characteristics of users and services over TCP/IP. LDAP is used in several centralized user modeling servers, such as [Kobsa and Fink, 2006] and [Yimam-Seid and Kobsa, 2003]

Web-based protocols. Nowadays, most of the applications use Internet protocols to transfer data. The simplest way for applications to communicate over Internet is to use the HTTP protocol, by means of a GET request and a reply in some format (such as XML or JSON).
In CUMULATE [Brusilovsky, 2004], the communication is based on a simple HTTP GET request - XML reply, and the messages to the student modeling server are sent in a form of GET requests as well. For long queries, also XML request - XML reply is allowed.
In DOPPELGANGER [Orwant, 1995] the clients and the server communicate through TCP/IP protocol.
In PersonisAD [Assad et al., 2007] the interaction of the applications with all the knowledge models (of people, sensors, device, places) is based on the implementation of simple operations: access (each entity and its associated model has a globally unique ID; the access operation is used to locate the model server and connect to it), tell (a piece of evidence -value, source, type- is added to a given model component within a given model context), ask (a value for a model context/component is returned after resolution by a resolver function). To support distributed models, these operations (and others for model management) are implemented using a simple remote procedure call mechanism based on JavaScript Object Notation (JSON) with HTTP as the transport protocol.
Grouplens [Konstan et al., 1997] comprises an associated set of APIs. Via this interface, applications can send ratings to, and receive predictions from the GroupLens recommendation engine. Since there was no standard protocol for exchanging ratings and predictions, they defined an open protocol for communication between news readers and the GroupLens server. To further simplify the task of caching data and following the protocol, they implemented and distributed client libraries written in C and Perl. The client libraries define a simple API that news readers can use to request predictions and to transmit ratings.

Otherwise, applications can exploit most complex architectures for communication, as the stacks WSDL/SOAP [W3C, 2001] infrastructure for Web Services. A Web service has an interface described in a machine-processable format (WSDL) and the other systems interact with it in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. There are several examples of systems that perform UM interoperability through the loosely coupled structure of Web Services. Among the systems we analysed, we can cite the following ones.
In MEDEA platform [Trella et al., 2003], all the requests and responses between MEDEA's services are managed by the Connection Manager module that communicates with the instructional resources using the description provided in the WSDL file.
In MUMS [Brooks et al., 2004], the interaction between the systems (user data producers and user modellers systems which receive such data) is mediated by a central broker component. The communication among the modelers and the broker is enabled either through a publish/subscribe model or a standard web service technology (query/response model).
In Musa and Palazzo Moreira de Oliveira [2005], the main element of the architecture is the repository which stores the learner model, with data collected from various e-learning systems. The communication between the repository and every learner system requires the system to be implemented as a web service.
In Zhang et al. [2006] the user model data are exchanged according to the user model schema that the service provider has registered in a UDDI registry [UDDI, 2004].
Cena and Furnari [2009] implement their interaction model in a framework built upon a Service Oriented Approach (SOA)-based environment, enriched with an enhanced discovery tool.
Notice that web services may be used both in centralized and decentralized settings, as in

Dolog and Shafer [2005a], that proposed a centralized approach where the user model fragments can be accessed by the use of a web service which acts as a learner model server. The model can be accessed directly by invoking the functions of a web service or in a synchronized way. In the latter case, each client maintains its own repository which has to be synchronized with the main server every time a change occurs. The proposed web services framework can be used in a distributed way as well, where several services exchange the learner models between each other.

5.2 Languages for the exchange of user model data

Besides how they communicate, different languages at a higher level can be exploited to model the interaction among providers and suppliers of user model data.

Systems may use a traditional *message-based language*, like ACL[13] and KQLM [14]. For example, in LSM [Machado et al., 1999], the communication between the LMS server and the clients applications is defined as a subset of the KQLM messages.

A language specifically defined for UM interoperability is *UserML* [Heckmann and Krüger, 2003], a RDF-based unified user model exchange language, which enables decentralized systems to support the share of user model data among different applications in a ubiquitous context. UserML includes the sublanguage UserQL, which defines the queries in URI format that can be sent via the "get method" to the server which stores the user model data. Examples of work using UserML for communication between applications are Heckmann et al [2005a] and Lorenz [2005][15].

Most of the proposed solutions are mainly conceived for simple interactions. However, in an open environment like the Web (especially in absence of a shared user model representation), there might be the need for clarifying and negotiating the user model data. To this purpose, Cena and Furnari [2009] propose to negotiate the content of a user model by exploiting ad hoc high-level communication protocols which regulate complex interactions. These protocols are expressed as Dialogue Game based on Speech Acts and can be represented by means of some XML-based declarative formats.

Another example of high level communication protocol specifically conceived for negotiating UM data is the *Cross-System Communication Protocol (CSCP)* proposed by Mehta [2008]. This protocol is responsible for managing the communication between the Context Passport[16] and any information system. CSCP first negotiates so that the user Context Passport and the system can agree on the information to be exchanged. The negotiation here is especially about agreeing on a vocabulary as a part of the negotiation.

We can conclude noticing that this dimension has been addressed by most of the analysed work, since this aspect is crucial for the real feasibility of the specific proposed approach. Only few work does not consider this issue at all (Niu et al., 2003, Carmagnola and

[13] The *ACL* - Agent Communication Language [ACL, 2002] allows to express interaction protocols for the message exchange and communicative acts.

[14] *KQLM* - Knowledge Query and Manipulation Language, a high-level communication language which allows to define a common format for messages.

[15] In Lorenz's framework, the architecture has been defined independently from the specific language for the inter-agent communication. However, the authors considered UserML a good candidate to be used.

[16] The Context Passport is an active component hosted in the browser toolbar of the client device, in charge of storing and extracting user information and supplying the relevant ones to the system, according to the current user's context of use.

Dimitrova, 2008, Berkovsky et al., 2008).
The solutions exploiting standard approaches to communicate are those having more opportunities to be adopted. For example exploiting web-based protocols/approaches (e.g. HTTP, JSON, WSDL, REST, etc.) may facilitate the feasibility of communication and the exchange of data. At the same time, all the work that provides APIs - which are now widely adopted on the web such as Google APIs - for different purposes (e.g. communication, integration) offers a more feasible approach to UM interoperability. On the other side, work which propose ad hoc solutions has the limitation to be less scalable (e.g. Cena and Furnari [2009], Mehta [2008]).

6 Exchanged data

This dimension deals with the type of data that can be exchanged during the UM interoperability process. In the work we have analyzed, the data that are exchanged are of different types: user data, usage data, environment data, domain data, inferred data, reasoning data and social data.

User data: according to the definition of Kobsa et al. [2001] user data may include demographic data, user knowledge, user skills and capabilities, user interests and preferences, user goals and plans, etc.
Usage data: according to Kobsa et al. [2001] usage data may include selective actions, temporal viewing behavior, ratings, purchases and purchase-related actions, usage frequency, situation-action correlations, actions sequences, etc. A kind of usage data that interoperable systems may exchange can be defined as *raw data*, namely uninterpreted data such as timestamp data, log files, actions history, etc., which have not yet been processed by the system.
Environment data: according to Kobsa et al. [2001], environment data may include software environment data, hardware environment data, local data, etc. In our classification, this dimension also includes all the data regarding the context.
Domain data: all the data involved in the domain of the application, such as domain concepts, the items managed by the applications (i.e., products, lectures, information, etc.) the material characterizing the domain (i.e., exercises, documents, videos, audios, etc.). Domain data are typically exchanged when the user model is an overlay [Brusilovsky, 1996] of the domain model.
Inferred data: data that have been elaborated by the system for modeling and adaptation purposes, such as elaborated user model data, interests predictions, clusters, stereotypical classifications, etc.
Reasoning data: data explaining where the derived information comes from (e.g. the reasoner used, the reasoning methods, the inference rules, the assumptions, etc.).
Social data: the social features of the user, such as her friends, community and groups membership, etc.

In the following table we present the work we analyzed with respect to the kinds of data they exchange, and then we discuss it.

	Kind of model	User data	Usage data	Environment data	Domain data	Inferred data	Reasoning data	Social data
Exchange of user/learner models								
Niu et al, 2003	User Model	X		X		X		
Lorenz, 2005	User Model	X	X	X		X		
Dolog et Shafer, 2005	Learner Model	X	X	X	X	X		
Musa and Palazzo Moreira de Oliveira, 2005	User Model	X	X			X	X	
Metha et al, 2005	User Model	X	X	X				X
Heckmann, 2005	User Model	X	X	X		X		
Zhang et al, 2006	User Model	X	X	X		X		X
Van der Sluijs and Houben, 2006	User Model	X	X	X		X		
Carmagnola and Dimitrova, 2008	User Model	X	X		X	X	X	
Berkovsky et al, 2008	User Model	X	X	X		X		X
Cena and Furnari, 2009	User Model	X			X			
User modeling/adaptation service								
Group Lens, 1997	User Model		X			X		
Learn Sesame, 1997	User Model	X	X	X		X		
PersonisAD, 2007	User, Place, Device models	X		X		X		
MUMS, 2004	Learner Model	X	X			X		
UMS, 2006	User Model	X	X					
Share user models across applications								
DOPPELANGER, 1995	User Model	X	X	X		X	X	
LMS, 1999	Learner Model	X	X		X	X		
MEDEA, 2005	Learner Model	X			X	X		
CUMULATE, 2004	Learner, Group Model	X	X		X	X		

Table 3 Exchanged data.

In DOPPELÄNGER [Orwant, 1995] the server is able to receive disparate data about the users and provide them after having interpreted them according to multiple learning techniques. Thus, it exchanges user data and inferred data.

Grouplens [Konstan et al., 1997] exchanges usage and inferred data. Specifically, Usenet clients send explicit and implicit user ratings to the server, which sends back interest predictions for the corresponding articles.

In Learn Sesame [Caglayan et al., 1997] relevant user and usage and environment characteristics are sent to the server in form of attribute-value pairs. Then these time-stamped events are analyzed and then clusterized, in order to discover recurrent patterns. Hence, inferred data are also exchanged.

In LMS [Machado et al., 1999], the learner model data, which are an overlay over the domain model, are exchanged. In particular, the learner model contains i) the declarative knowledge level associated with a particular topic, that is system's inferences about the user level of knowledge, and ii) the information about some actions performed by the user on the topic, namely coarse data not elaborated by the system.

The PersonisAD [Assad et al., 2007] framework supports the use of models by arbitrary applications such as those within a pervasive computing environment. Such applications can add evidence to the components in the models or access the value of sets of components in the models. This can include the model of people (containing information about user's location, and user knowledge, preferences and goals), as well as the models of places, sensors, services and devices (although these typically have highly restricted access). Hence, user and environment data are exchanged. The model is also updated, or elaborated, by means of the addition of evidence when rules on components fire or when an authorized application adds evidence. Hence, inferred data can also be exchanged across applications.

In the MEDEA platform [Trella et al., 2003], systems exchange the learner model and the related domain model. In particular, the student's model in MEDEA is composed of the Student Knowledge Model (what the student knows about the subject), and of the Student Attitude Model (other student features which are relevant for the instructional process). The Student Knowledge Model is an overlay model divided into i) the estimated model (what the system guesses about the student's knowledge based on the student's behaviour), ii) the verified model, with the data obtained from the evaluative components; iii) the inferred model from the prerequisite relationships; and iv) the inferred model from part-of relationships. Hence, inferred data can also be exchanged.

In Niu et al. [2003] all the available user information is exchanged. Since the data can be collected in different contexts and for different purposes, the systems also have to exchange the information about the context and purpose for which the data was collected, to allow for the integration and reuse of the data in the the new context and for the new purpose. Hence, user data, environment data and inferred data are exchanged.

In MUMS [Brooks et al., 2004], the authors adopt the definition of opinions to represent the learner data exchanged across systems. An opinion is defined as a temporally grounded codification of a fact about a set of users from the perspective of a given event producer. Opinions include direct observations of the interaction between the learner and the system, beliefs about the learners knowledge, desires, and intentions. Moreover, every opinion transferred within the MUMS system includes the timestamp indicating when it was generated. The system is used to exchange user and usage data.

In CUMULATE [Brusilovsky, 2004] the servers send the student model servers a set of events. Each event specifies the activity server, the kind of event, the activity producing the event, and an outcome of the event. External and internal inference agents process the flow of events in different ways and update the value in the form of pairs (property-value) and triples (property-object-value). Each inference agent is responsible for maintaining a specific property in the inference model, such as the motivation of the student, the current level of knowledge, etc.

In Lorenz [2005] a lot of information about the user can be exchanged across agents (user, usage and environment data). Indeed, the user model the agents refer to represents different characteristics of the user (basic user data, preferences, goal and task information, etc.) and the context of use (physical and environment data). Also inferred data are exchanged.

In Dolog and Schafer [2005a] learner model data are exchanged. These include the features contained in the most used learning model specifications such as PAPI and LIP. The main features are: learner performance, competencies, and preferences (language, device); learning style, certificates, evaluation and assessments. The system also exchanges the so-called portfolio, which contains other educational activities, materials, lessons, and projects created within the activities. User, usage and environment domain data, as well as domain and inferred data, are exchanged.

In Musa and Palazzo Moreira de Oliveira [2005] a learner model based on IEEE PAPI and LIP standards is exchanged. The learner model contains several categories of data about the user, like personal data, relations information, security information, preference information and data about the performance. Moreover, such basic information is enriched with additional definitions of learning styles and cognitive styles. Thus, user, usage and inferred data are exchanged.

Mehta et al. [2005] several features of the user and her situation can be exchanged: cognitive characteristics of the user (area-of-interest, competence, preference) (user data), task dimension (current task, task role, task history) (usage data), relation dimension (relationships the user is involved in) (social data), environment dimension (device, current time, language, location) (environment data).

In Heckmann [2005b] the exchanged user data includes basic user dimensions, such as demographic data, user knowledge, emotional state and personality aspects, user skills, capabilities, user interests, preferences, user goals and plans, etc. Moreover, the intelligent environment can also be modeled and exchanged by using data like location, time, device, etc. Thus, user, usage, environment and inferred data are exchanged.

In Zhang et al. [2006] the authors move from the idea that the exchange of user models could involve services from various domains interacting with users having different roles. In particular every role is described by several dimensions: *Geninfo* (user id, location, email, phone number, etc.), *Preference* (habit, competence and general interest of user), *Competence* (skill and expertise), *Privpreference* (privacy preference), *Geninterests* (user interest), *Relationship* (social relationships the user is involved in), *Task* (goals, history task and current state) and *Taskrole* (concrete role of the user in a specific task). Thus, user, usage, environment, inferred and social data are exchanged.

UMS [Kobsa and Fink, 2006] allows for the exchange of user data (demographic data, interest and preferences) and usage-related data.

In GUC [van der Sluijs and Houben, 2006] any type of user data is exchanged, from domain-independent user data (such as demographic features) to domain-dependent data, such as preferences, behavior, knowledge and all the other facts that are considered relevant for the application. Thus, user, usage, environment and inferred data are exchanged.

Carmagnola and Dimitrova [2008] focus on the exchange of those user data that can be represented as an overlay model based on the domain model, such as knowledge, preferences and interests. Thus, user, usage, domain and inferred data are exchanged.

In Berkovsky et al. [2008, 2009] systems exchange user data for various aspects of the user, an item, the context and the evaluation of the personalization features. In particular, the aim is sharing the experiences of a user for an item. An experience is defined as an evaluation function that maps a triple (that is, the user who experienced the experience, the item experienced by the user and the context) to an evaluation. Thus, user, usage, environment and inferred data are exchanged. Also social data are exchanged. For example, in cross-domain mediation, collaborative recommenders may exchange IDs of similar users. So, if a system needs to build the neighborhood of similar users, it considers users who were similar to the target user in other domains. It may not be the standard social data coming from social networks, but it can be considered as some form of community knowledge.

In Cena and Furnari [2009] approach, systems can exchange user data, both domain-independent features (such as demographic features) and domain-dependent features (in particular, the ones defined in GUMO, such as interest, knowledge, preference). In the latter case, the domain data is exchanged too.

As the reader can notice, user data, usage data, environment data, and domain data are frequently exchanged by almost all these systems. Less attention is devoted to domain data, reasoning data, and social data. However, we believe that, when possible, it could be useful considering also the exchange of these information. For instance, the exchange of domain data as domain ontology could be used to enhance the interoperability process with the addition of this semantic component. The exchange of reasoning data could be useful, for instance, both for the interoperability of user modeling strategies (see Section 11) and when used as provenance information (i.e., meta information representing both proofs and proof metadata) with the aim of providing explanations about the reasoning processes (see the Inference Web Initiative[17]). The presence of social data on the web is an increasing phenomenon, thus we may foresee an increase in presence of social data in future interoperable systems. Finally, we notice that most systems do not exchange raw data, which could be useful when a system wants to analyze user actions made on other systems, and then performing its own inferences.

7 Representation of the exchanged data

This dimension describes how the exchanged user models are represented from a syntactic and semantic perspective. The management of syntactic and semantic interoperability of user models has been variously addressed in the user modeling community: i) using a *common UM representation*, that is adopting the the same representation structure and languages to express the user models; and ii) using a *translation* approach, that is representing the user model with the desired structure and language, which then requires the data to be exchanged only after an agreement over the semantics of the data had been reached.

7.1 Common UM representation

To adopt a *common UM representation* among systems means that the user models of different systems should be expressed in the same representation structure and languages. Of course, when the user models of different systems rely on a common user/domain model, they can easily be exchanged and consistently interpreted.

[17] http://iw.stanford.edu/2.0/

	Common UM approach	Translation approach
Exchange of user/learner models		
Niu et al, 2003		X
Lorenz, 2005	X	
Dolog et Shafer, 2005	X	
Musa and Palazzo Moreira de Oliveira, 2005	X	
Metha et al, 2005	X	
Heckmann, 2005	X	
Zhang et al, 2006	X	
Van der Sluijs and Houben, 2006		X
Carmagnola and Dimitrova, 2008		X
Berkovsky et al, 2008		X
Cena and Furnari, 2009	X	
User modeling/adaptation service		
Group Lens, 1997	X	
Learn Sesame, 1997	X	
PersonisAD, 2007	X	
MUMS, 2004	X	
UMS, 2006	X	
Share user models across applications		
DOPPELANGER, 1995	X	
LMS, 1999	X	
MEDEA, 2003		X
CUMULATE, 2004	X	*

Table 4 Representation of the exchanged data.

(*)In a further extension of the system, a translation approach has been developed, so that the system can take events in any ontology through the translation service [Yudelson et al., 2007].

In DOPPELÄNGER [Orwant, 1995] the user models are encoded in a simple knowledge representation language called SPONGE, which utilizes LISP-like data structures manipulated by C and Perl programs.

In Grouplens [Konstan et al., 1997] applications that intend to communicate a *priori* user data to GroupLens have to convert them into the format accepted by GroupLens and store them in its ratings database.

In the Learn Sesame server [Caglayan et al., 1997] the domain is modeled by means of a Model Definition Language (MDL) that has to be used both by the applications that want to communicate with the server and by Learn Sesame server itself.

In LMS [Machado et al., 1999] the representation of the learner profiles and of the domain knowledge is based on standards vocabularies, such as IEEE, PAPI, IMS, RDCEO, and LIP. Applications should convert their learner model as well as their domain model according to a predefined structure.

In CUMULATE [Brusilovsky, 2004] there is a single domain model for each subject and all user and usage data should be sent using this model [Brusilovsky, 2004, Yudelson et al., 2007].

Musa and Palazzo Moreira de Oliveira [2005] propose a framework where the learner model relies on accepted data models, that is, PAPI and LIP standards. Indeed, the exchanged data are represented in XML and follow the PAPI_LIP model. To convert the data for the participants systems database, a wrapper is required. On the contrary, if the database used by the systems stores the data in XML following the PAPI_LIP format, the wrapper is not necessary.

In UMS [Kobsa and Fink, 2006] the basic unit of information in an LDAP directory is encoded in LDIF - LDAP Data Interchange Format.

Recent work has exploited Semantic Web [Berners-Lee et al., 2001] standards and technologies to enable syntactic and semantic interoperability. Many solutions have employed Semantic Web standard languages, such as RDF[18], RDFS[19] and OWL[20] for representing the user models. Related to the Semantic Web is the adoption of the common UM representation which is, very often, a shared *user model ontology* for representing the user (and, if required, her current context and the domain).

Among the analyzed work, many have exploited shared user and domain ontologies. In MUMS [Brooks et al., 2004], the opinions about a user, the information indicating when the opinions have been generated, and the administrative information about the system that provides such data are expressed in RDF. To foster interoperability, a separate ontology database is maintained. This database is not authorative, for example if an ontology is missing in the database the opinions are still passed around in MUMS. It is expected that the consumers of the opinions will understand their semantics. So new ontologies do not affect the deployment of the MUMS system. Authors can inspect such a database to understand the semantics of the collected data and to support the reuse of previously deployed ontologies.

In Dolog and Schafer [2005a] the learner model is represented by means of a learner profile ontology expressed in RDF, configured from fragments based on three specifications: IEEE PAPI, IMS RDCEO and LIP.

In Mehta et al [2005] the user and her situation are modeled by using an extensible Unified User Context Model (UUMC) ontology. UUMC is published as a shared ontology all the participating systems can rely on. The user profile information represented as facets and dimensions of the UUMC ontology are encapsulated in a context passport (in the form of a browser toolbar) that accompanies the user when traveling through the information space. When the user performs an activity within an information system, her context passport is presented to the information system. Since the context passport is bound to the UUMC shared ontology, the system is able to partially interpret the context passport using a mediator architecture.

In PersonisAD [Assad et al., 2007], the application using a Personis model has to know the name of the components and context. Thus, in that sense, there is the need for shared knowledge at an ontological level of the component.

Zhang et al. [2006] represent the user models and their related roles with a shared ontology, that is, the User Role Model Ontology. Such an ontology can be extended according to the features of the service provider and on the bases of the information it requires.

Heckmann [2005b] introduced the General User Model Ontology (GUMO) to represent user data. According to the author's vision, all the user-related data can be shared across systems if they are represented through the GUMO ontology. The user features represented in GUMO are defined at a high level in order to be employed by as many systems as possible. Moreover, GUMO can be extended by other existing ontologies. In this direction, the author extended GUMO using the Ubisworld knowledge-base[21] which introduces classes to model many contextual characteristics of a user, including their activities, as well as the environmental context. It also provides a symbolic spatial model to express location. GUMO is also used in Lorenz [2005] and in Cena and Furnari [2009] approches .

18 www.w3.org/RDF/
19 http://www.w3.org/TR/rdf-schema/
20 http://www.w3.org/TR/owl-features/
21 http://www.ubisworld.org

7.2 Translation approach

Even if exploiting a common user model representation is promising, in the user modeling field the use of common ontologies is still far to be widely adopted. On the one hand, even if the designers of user modeling systems more and more frequently choose to represent the user and domain models as ontologies, they tend to employ their own ontologies. On the other hand, in an open environment like the Web it is not always feasible for applications to rely on a pre-defined user model representation.
This has led to an opposite approach for representing user data in a interoperability context, where applications can represent the user model as they wish, and user model data can be exchanged only after an agreement over the semantics of the data had been reached (*translation approach*). This implies the general task of models alignment and the resolution of multiple conflicts on different levels. Specifically, any conflict at the language level (such as different syntax) needs to be solved, as well as any difference at the semantic level (such as different structure and semantics among models). Sosnovsky et al [2009] proposed a classification of the possible conflicts occurring when dealing with multiple user and domain models: i) naming conflicts (the same concept is defined in two models by different terms or the same term defines different concepts); ii) different graph structures (the models choose to connect relevant sets of concepts in different ways); iii) different scopes (two models cover parts of the domain that only partially intersect or the scope of one model includes that of another model); iv) different granularity (the size of concepts differ across the models; a single concept of one model represents a piece of domain knowledge covered by several concepts in another model); v) different focus (the models make use of different modeling paradigms or adhere to different modeling conventions).

In the *translation approach* every system stores its own user and domain models, while UM interoperability is achieved by the creation of a central model which serves as a shared storage containing the most used concepts within the domain. In order to exchange user model data, each application is expected to map to the shared user model. Such a mapping requires a *translation (or mediation)* of the user models. This process can be defined as "importing the user modeling data collected by other (remote) personalization systems, aggregating[22] them and generating an integrated user model for a specific goal within a specific context" [Sheikh and Sticklen, 2002]. The translation of the user models is typically managed by a *mediator* component. For example, a mediator component is adopted and described in [Berkovsky, 2006]. In this framework each application has its own user model. However, every user model has to be converted into a shared-user model expressed with an ontology written in a Semantic Web language. Since each service provider stores a partial user model in its own format and representation, the mediator component is in charge of mapping both the syntax and semantics from each system into the syntax and semantics accepted by the shared user model. In detail, the mediator is responsible for the following tasks: *i)* mapping from specific services to a generic representation and vice versa, *ii)* providing standard language/interface for the exchange of user model data, *iii)* maintaining user modeling semantic knowledge for facilitating *ad-hoc* mapping.

Other work that exploits the translation of the user models through a mediator component has been presented in Van der Sluijs and Houben [2006]. In this approach, each application can represent user data with its own vocabulary which can be expressed in a proprietary format. User model data are exchanged across systems through a mediator component called

[22] In this definition, the term aggregation implies resolving the heterogeneities and inconsistencies in the obtained data (see Section 8).

GUC -Generic User Model Component. In the GUC component the translation exploits a data integration engine which merges different models into one by means of mapping rules. The mappings are performed semi-automatically using the SPARQL language. However, the framework adopts a hybrid approach since it also enables direct mapping among the applications when a more custom-made exchange of application-specific information is needed.

Other approaches manage the conversion between the models by exploiting automatic or semi-automatic ontologies mapping techniques [Doan et al., 2003], such as Carmagnola and Dimitrova [2008] , where distributed user models are handled even in the absence of a unique user model data representation. The user and the domain model must be represented as RDFS ontologies and the conversion among user model is performed by exploiting automatic mapping techniques based on string metrics. Also in Cena and Furnari [2009] systems are not compelled to share a common domain model. The translation among the user models is performed by means of a dialogue game, which can be used to exchange and negotiate concepts belonging to different ontologies.

Finally, when the knowledge models (user and domain model) are not represented by using Semantic Web technology, reaching a shared syntax and semantics has been made possible by using *machine learning* techniques. These techniques provide a promising alternative by using example data to learn mappings between profile formats without the need to rely on accepted semantic standards or ontologies [Mehta, 2007].

To conclude we must notice that the success of the user model interoperability process in term of representation of the exchanged data depends on several factors. First, it relies on the creation of an agreed upon ontology for the exchanged data which various applications may use during the interoperability process, for example ontology like GUMO, UUCM or standard vocabulary like PAPI, LIP, etc. Moreover using standard web languages (e.g., XML-based languages like RDFS, OWL, etc.) may help in using/re-using, exchanging, and interpreting the data. Then, the success of the user model interoperability process also relies on the adaptation purposes of the applications involved in the process and on the benevolent contribution of developers writing user-adaptation code to create ad hoc procedures for different purposes. The contributed procedures could become a library of adaptation software, that can be shared, in open-source fashion.

Concerning the limitations of the proposed approaches, for many authors the biggest limitation is the reliance on the creation of an agreed upon ontology of user data types, which various applications may store. For some authors, there is not the need to have a full ontology, but in order to find user data fragments scattered around the world, they require some kind of dictionary, mapping, and generally some semantic agreement, for example see Niu et al. [2003] and Assad et al. [2007]. Specific approaches may need specific ontologies, for example an ontology of purposes is envisaged by Niu et al. [2003], and an ontology of context by Assad et al. [2007]. In the first case, there is the necessity of the creation of a taxonomy or even ontology of adaptation purposes, which represents meaningful relations between purposes. In the latter case, PersonisAD assumed that the user model is organized in contexts, where each context defines a semantic space. This means that applications that need to interoperate should agree on the semantic model in a particular PersonisAD context.

On the other hand, adopting a translation paradigm rather than a joint vocabulary has advantages since every application can use its own UM data structure, but this also has disadvantages in the sense that every pair of application that want to exchange UM data need to create a mapping in both the directions. The main limitation is that the mediator has to implement many conversion mechanisms for different forms and domains of UMs obtained from the service providers. This is also a scalability limitation. If a new UM representation is introduced, the mediator would need to develop new conversion mechanisms from this representation to all other representations and vice versa. For complex UM data structures this is not a trivial task, even though authors of these mappings do get tool support providing suggestions of potentially good mappings; see for example Berkovsky et al. [2008], van der Sluijs and Houben [2006], and Dolog and Schäfer [2005a] .

8 Integration of the exchanged data

This dimension describes how the user model data collected during the interoperability process can be integrated by systems, and how systems deal with possible conflicts. We can distinguish two approaches: i) the user model data collected by the different systems are not merged into an existing knowledge structure, but they are computed just in time as it is needed; ii) the collected data are merged into an existing knowledge structure. This requires a system to perform operations of "data integration" and to solve possible conflicts which may occur, with particular regard to data model and data value conflicts.

8.1 Not-merging the collected data

In some of the work we analyzed [Niu et al., 2003, Lorenz, 2005, Berkovsky et al., 2008, Assad et al., 2007, van der Sluijs and Houben, 2006] the user model data collected by the different systems are not merged into an existing knowledge structure and applications can use the data as they are, without requiring any integration. This is the case of *dynamic user modeling* which refers to the collection, on demand, of data about a user whatever user information is available at this moment from various agents and interpreting it for a particular purpose [Niu et al., 2003]. In dynamic user modeling there is not a single user model but a virtual infinity of models about an individual. Indeed, no integration of data is needed, since data are computed "just in time" by computational agents for a specific purpose.

In Niu et al. [2003], the user model data gathered by the different systems are computed just in time by the system that needs to perform adaptation, instead of being merged into a knowledge structure. Specifically, a purpose-procedure, retrieved from a library of possible purpose procedures is adopted. The procedure defines how to integrate the user data received from different systems depending on the context and purpose for which the system have collected the data. The purposes can be organized into hierarchical layers which allow to execute their respective procedures in an incremental manner.

In Lorenz [2005] there is no an explicit integration of the user model fragments collected by the different systems. The overall user model emerges from the information that a provider puts at the disposal in the current context. This approach enables highly dynamic models, which can even change in structure, if the context of the user changes for example because of moving to another technical environment.

In Berkovsky [2008] the exchanged data are not merged. The data are stored by the service providers and transferred to the mediator upon request. In the mediator these are converted into a shape and form that was requested (there is not predefined structure) and not stored by the mediator. So, the integration happens on the fly when the data are needed.

Also in GUC [2006] and in PersonisAD [2007] there is no merge of the collected data.

8.2 Merging the collected data

Other sytems, instead, merge the collected data into their user model. This requires them to perform operations of “data integration”, which involves the process of merging the partial user model data collected by different user-adaptive systems into a user model of a proper granularity. For example, a mapping procedure for the integration is presented also in the work of Dolog and Schafer [2005a, 2005b]. The procedure is implemented in the API available.

When the UM interoperability process leads to the integration of the user data collected from one or more participant systems, possible conflicts may occur at different levels: at the data model level and at the data values level. If the interoperable systems share the same vocabulary and the data schema (see Section 7.1) the conflict at the data model level is unlikely to happen. On the contrary, if systems exploit diverse internal vocabularies and data models (see Section 7.2), they need to map the data model of the collected user data with their internal one. This can cause some conflicts. We address these conflicts as *conflicts at the data model level*.

Some solutions have been proposed to manage the mapping of the data across applications in order to solve possible conflicts at the data model level (translation approach, see Section 7.2). Independently by the sharing of a consistent data model among systems, a further type of conflict when merging different data involves the data values. Indeed, different systems may have contradictory data values for the same user feature. For instance, system A may assume Mary’s “interest in art” is low, while system B may assume her “interest in art” is high. We address this conflict as *conflict at the data value level*.

The solutions which have been proposed in the work we analyzed mainly exploit the usage of meta-information and heuristic rules. Heckmann [2005] proposed the exploitation of software modules namely Conflict Resolvers, that is special kinds of filters which control the conflict resolution process. The working of the Conflict Resolvers is based on conflict resolution strategies with situational conflict categories, comparable to Martin [2007] and Ram and Park [2004]. The retrieval mechanism embeds a multi-level conflict resolution strategies which resolve queries according to common sense heuristics applied to meta information.
Another solution was proposed by Carmagnola and Cena [2006b]. In case of conflicting data values, systems consider those data having the highest reliability. To provide systems with the means of measuring the reliability of a user model data, the authors introduce a set of semantic meta-information mainly about how the data has been derived. Then, that meta-information is processed with a set of heuristics rules to derive a reliability measure.
A similar approach can be found in Carmagnola and Dimitrova [2008]. In this approach, the conflicts among data values coming from different systems is handled by measuring the credibility of every user model statement. Indeed, moving from the theory of evidence [Kadane and Schum, 1996], the system which retrieves the user model data from a provider computes the credibility of both the data and of the provider of the data. Credibility is assessed by exploiting a description of the context of the provider system as well as a description of every piece of user model data. In the case of user model conflicting values, the conflicts are solved by considering the user model data having the highest credibility value.

As a particular case, we must cite the work of Niu et al. [2003], where both the approaches are possible. It entirely depends on the purpose. For some purposes the data does not need to be merged, but adaptation decisions can be made based on the received data. Other purposes will use the data obtained from different sources to compute new user modeling data and this merged data can be either stored (and then sent to other places and purposes) or merged with other data received from other places to make a decision.

As a final consideration, we can notice that several work [Orwant, 1995, Konstan et al., 1997, Caglayan et al., 1997, Machado et al., 1999, Trella et al., 2003, Brooks et al., 2004, Brusilovsky, 2004, Musa and de Oliveira, 2005, Mehta et al., 2005, Kobsa and Fink, 2006, Zhang et al., 2006, Yudelson et al., 2007, Cena and Furnari, 2009] has not investigated the integration of the data exchanged during the interoperability process or has not proposed any solution to manage possible conflicts among the data.
This can be explained by the fact that very few work are really "'working systems"' that perform the UM interoperability process from the beginning to the end. What happens during the data exchange phase is considered as a problem of the system that collects the data, thus the provider of the data very often does not consider this issue.
However, UM interoperability can be really useful only when the exchanged data are effectively employed into cooperating user-adaptive systems. Indeed, managing all the possible conflicts which may occur when merging different partial user model data into another user model is a crucial task that needs to be taken into account. Thus, offering solutions for conflicts managing could promote the UM interoperability process.

9 Privacy

This dimension addresses the question of how to manage the privacy issues in the context of UM interoperability. In general, the main requirements for personalisation in relation to privacy are to obtain permission from the user to collect and use her data, and to protect the collected data. Furthermore, in an interoperability context, systems have to deal with the release of user model data to third party systems [Kay and Kummerfeld, 2006]. The privacy issues to be considered for UM interoperability are the following [Wörndl and Koch, 2003, Brar and Kay, 2004, Kay and Kummerfeld, 2006]:

- to protect user data by using adequate security measures such as anonymization, encryption, selective access [Schreck, 2003]. This issue deals with security, which is not a goal in itself but an auxiliary means to ensure privacy [Brar and Kay, 2004];
- to make the user aware of the fact that her data will be exchanged with the third parties, even if this is in contradiction with the seamless collaboration among systems foreseen in the ubiquitous computing vision;
- to manage the fact that applications may use data sources for other purposes or in another way than originally intended [Wang and Kobsa, 2007]. Thus, the user may give her permission to use data for a different purpose from the original;
- to choose which applications are allowed to import data, defining different access levels [Kobsa, 2007a] according to the trustworthiness of the applications. This implies some mechanism for the evaluation of the trust in the systems;
- to decide which data to share: for example, a user can refuse to reveal sensitive data. This implies the understanding of the most valuable portions of the user model for the adaptation goals and releasing only those portions [Berkovsky et al., 2006b];
- to deal with specific legal requirements which may affect the possibility of sharing user model data in an interoperable context [Kobsa, 2007a]. For example, a user model server may be forbidden to supply data to other systems if they use the information for different purposes with respect to the starting application;
- to deal with the issue of the ownership of the data, that is, if the data and the conclusions drawn from the analysis of the data belong to the users or to the system which collects

them. The trend is to consider the user as the owner of all the personal information stored in the profile, including the name, the address, and the areas of interests[23].

In the following table, we describe the solutions and approaches proposed and adopted by the work we analyzed. Such solutions can be grouped into: i) different access rights, ii) pseudonymous personalization, iii) encryption techniques, iv) perturbation techniques, v) scrutable user model, vi) joining consortia and organizations.

[23] "A method for guaranteeing privacy is to ensure that the user retains not only the computational ownership of the data [..] but physical ownership as well. For example, DOPPELGANGER can store user model on a PCMCIA card as well on the disk"[Orwant, 1995].

	Different access right	Encryption	Perturbation	Scrutable user model	Third party guarantors and standards	Pseudonymous personalisation
Exchange of user/learner models						
Niu et al, 2003	X					
Lorenz, 2005						
Dolog et Shafer, 2005	X					
Musa and Palazzo Moreira de Oliveira, 2005						
Metha et al, 2005		X				
Heckmann, 2005				X		
Zhang et al, 2006						
Van der Sluijs and Houben, 2006				X		
Carmagnola and Dimitrova, 2008						
Berkovsky et al, 2008			X			
Cena and Furnari, 2009						
User modeling/adaptation service						
Group Lens, 1997					X	X
Learn Sesame, 1997					X	
PersonisAD, 2007	X			X		
MUMS, 2004						
UMS, 2006					X	
Share user models across applications						
DOPPELANGER, 1995	X					
LMS, 1999						
MEDEA, 2003						
CUMULATE, 2004						

Table 5 Privacy.

Different access rights. The simplest way to protect the data is to *assign different access rights* to the services. In centralized user models servers, it is possible to discriminate different levels of access through the use of a "role-based access control". Only systems that are authorized by their *role* [Kobsa and Schreck, 2003] can access the server. Moreover, systems can have different levels of freedom in importing the data according to the privacy policies established by the user.

A variant of this approach applies different access rights at multiple levels of the model. This approach is the one adopted in Personis [Kay et al., 2002]. First, Personis supports restricted access at the component level. Second, filters are associated with applications so that only certain classes of evidence are allowed for each application. So, for example, one application may be allowed all the evidence, while another may only be allowed evidence that was given explicitly by the user and it is not allowed any other type of evidence, such as that based on observations of the user. The filters can also operate at the level of the evidence source, so that, for example, one application may be allowed only the evidence that came from a movie recommender application. The third level of privacy control is available through the resolver since an application is only allowed access to the model via a particular set of resolvers. These can make arbitrary interpretations of the data. For example, in PersonisAD, the Point Query [Assad et al., 2007] is only allowed the value defined by the last piece of evidence. On work on location privacy, the resolver can be restricted to evidence from work hours over the last day.

Also in Dolog and Shafer [2005a] there is a privacy dimension implemented in the API as well as in the model, according to standard model they have considered. This is the base for giving different access right to the applications which want to access the user model data.

In Niu et al. [2003], even if the problem of privacy is not addressed explicitly, the idea is close to different access rights. The idea is to have networks of trusted application that would allow sharing of data only by other trusted agents/applications [Niu et al., 2004].

Pseudonymous personalization. Another efficient way to protect data is the *pseudonymous personalization* [Kobsa and Schreck, 2003], that is, denying the access to the relationship between the user and her personal data. A user can use a pseudonym that is a unique and persistent identifier that differentiates her from all the other users. The user remains unidentifiable while maintaining a persistent identity across the sessions with the personalized system. In GroupLens [Konstan et al., 1997] the model of privacy was primarily about not requiring actual user identification, but just using a user-selected pseudonym.

Encryption. Other privacy-protected approaches are aimed at making it impossible to link the data back to the corresponding user. Along this line, a privacy preserving distributed framework was proposed by Mehta [2007]. In this approach, the user model data exchange and the contributions from each user are made available only in an encrypted format. More specifically, the author introduces a distributed version of probabilistic latent semantic analisys (PLSA), which can be used in a privacy preserving manner.

Perturbation techniques. In case of collaborative filtering-based systems, widely used techniques for preserving privacy are the so-called *perturbation* (sometimes referred to as *obfuscation*) techniques, consisting of some changes in the values of the exchanged user model. In this way, in the case of malicious attacks, the data will not reflect the exact content of the user profile. For example, in Berkovsky et al. [2005] a certain percentage of user's ratings are replaced by different values before the ratings are submitted to a central server for collaborative filtering[24]. In the same way, systems may dynamically provide vague answers

[24] Some researches [Berkovsky et al., 2005] showed that perturbation of user model data does not considerably reduce the accuracy of system's predictions and thus of the adaptation results.

or "white lies" to preserve user information.
Scrutable user model. An emerging issue regards the involvement of the user in the privacy management [Cranor, 2003, LaRose et al., 2004]. A possible solution is to provide users with the power of directly controlling privacy, using a *scrutable user model* technique which enables the user to scrutinize and modify any part of the user model and associate it with the preferences about the privacy [Kay, 2000]. In Personis [Kay et al., 2002] the user needs to explicitly allow an application to access relevant views in the user model. For instance, the user may decide that the security of a system is effective enough to allow it access to substantial amounts of the user model, or can specify that an application is authorized to access only a limited part of the user model.
In Heckmann [2005] the user can inspect, change, delete and control her personal data stored on distributed systems, devices or user model repositories by editing a specialized online editor (*AccessedEditor*), a web browser application that incorporates the privacy dimension. Similarly in GUC [van der Sluijs and Houben, 2006] users can control their privacy management via the user model editor which allows them to access all their stored data and to control which applications may access them.
Standard consortia and organizations. Finally, several applications (especially commercial ones) have joined standard, third-party independent consortia and organizations in order to ensure privacy for their users. For example, GroupLens [Konstan et al., 1997] was a member of several privacy consortia, such as TRUSTe[25]. Also Learn Sesame [Caglayan et al., 1997] participates in the P3P[26] project of the World Wide Web Consortium (W3C).
In UMS [Kobsa and Fink, 2006], user's privacy is ensured by the use of the LDAP directory, which gives a security model to provide standardized support for the security of the information and users' privacy. This is supported by facilities for authentication, signing, encryption, access control, auditing, and resource control.
DOPPELGANGER [Orwant, 1995] use a secure PGP-based protocol. Pretty Good Privacy (PGP) is a software that provides cryptographic privacy and authentication. PGP is often used for signing, encrypting and decrypting e-mails to increase the security of e-mail communications. Users have been willing to accept certificates and check their validity manually or to simply accept them. PGP follow the OpenPGP standard (RFC 4880) for encrypting and decrypting data.

We can conclude noticing that this dimension, even if crucial for the user model interoperability process, has not been considered by most of the analysed work [LMS, 1999, Medea, 1999, CUMULATE, 1999, MUMS 2004, Lorenz 2005, Musa and Palazzo Moreira de Oliveira, 2005, Zhang et al 2009, Cena and Furnari, 2009, Carmagnola and Dimitrova, 2009]. This could be motivated by the fact that there is not a standard solution adopted by the community that facilitates the solution of privacy problems.
Moreover, we can notice that also the work which consider privacy issue provides simple solutions that do not offers a real protection of data (like [Orwant, 1995] and [Kobsa and Fink, 2006] do). Indeed, most of the work simply gives the user the control over the scrutable user model (e.g. Heckmann [2005], [van der Sluijs and Houben, 2006]), without paying attention to other privacy-related issues (such as data protection). The adoption of standard like

[25] http://www.truste.org/
[26] The Platform for Privacy Preferences Project (P3P) enables web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents will allow users to be informed of site practices (in both machine- and human-readable formats) and to automate decision-making based on these practices when appropriate. Thus, users are not required to read the privacy policies at every site they visit. http://www.w3.org/P3P/

PGP-based (or similar) protocols could help in this sense, since they are easy to use and they support different levels of data protection.

10 Other UM interoperability-related issues

In this survey, we have extensively discussed those issues concerning the process of exchanging user model data. However, the issues pertaining to what comes “before” and “after” the data exchange phase have not been investigated. In particular, in the very first phase, interoperable systems have to discover other system(s) collecting the data about a specific user (service discovery - Section 10.1). Moreover, they have to agree on the identity of the user whose data are exchanged (user identification - Section 10.2). Then, after the exchange, applications should evaluate the reliability of the exchanged data (Section 10.3).
For completeness, in this section we briefly discuss these issues.

10.1 Service discovery

In general, discovery is “the process by which an entity on a network is spontaneously notified of the availability of desirable services or devices on the network” [Edwards, 2006]. In the UM interoperability context, service discovery is the process aimed at finding the systems storing the required user data. Most of the analyzed systems consider such an issue as a starting assumption.

From a technical perspective, there are different modalities to perform service discovery. Systems may use a peer-to-peer approach to find other systems. In this approach, every system is in charge of discovering the other systems holding the desired user data, as in Niu et al. [2003] and in EDUTELLA [Dolog and Vassileva, 2005], which use Gnutella peer-to-peer protocol for discovering user model data residing at different systems/agents.
The PersonisAD framework [Assad et al., 2007] provides a discovery of models, based on a DNS-based service discovery. This facilitates distributing models across various machines across the network.

Alternatively, a server can distribute a query over the entire network, and only the systems storing the requested value will answer, such as in CUMULATE [Brusilovsky, 2004]. Another possible solution is to exploit some kind of *registries* with “yellow pages” functionalities, like UDDI [UDDI, 2004] and MAS registry like Directory Facilitator [FIPA, 2002], where systems subscribe themselves providing some description about their services. All the analyzed systems that exchange user model data as web services [Brooks et al., 2004, Musa and de Oliveira, 2005, Zhang et al., 2006] exploit UDDI registries as discovery agencies. In addition, Cena and Furnari [2009] enhance the UDDI registry with the definition of the specific tools needed for implementing richer communications (for example dialogue schemas).

Another approach is to exploit some form of *discovery systems* [Guttman, 2006], that is, systems in charge of finding resources on the web, also called *matchmakers*[27]. For example, the Broker-based Discovery Service for User Models [Chepegin et al., 2005] allows various applications to discover and invoke semantically described user models. New applications register themselves with the User Model-Broker and provide it with the ontological description of their internal user model, domain and possible other application models.

[27] A matchmaker is a search engine a system can delegate to find services. It works by performing a match between the user request and the service description typically advertised on some public registry.

A similar solution is adopted by Lorenz [2005], where each component of the framework announces to the registered listeners the information it delivers and registers as a listener for all the information it needs. If one of the components fires an event, the event-message is sent to its local broker, which forwards the message to all surrounding registered brokers.

10.2 User identification

User identification is a relevant aspect of the UM interoperability process, since data sharing is subordinate to the correct identification of the user whose data are exchanged among systems. In spite of its relevance, the issue of user identification has been not tackled in most of the analysed work. Only few solutions have been proposed. We can mention the solution proposed by Dolog [2004] where the user identification is performed as unification of "identification records" maintained on the different systems the learner interacts with. Systems are allowed to use their local user identification schema and the mapping between the schemes is performed by a personal learning assistant which is in charge of retrieving all the instances of the identification concept for the current user in the identification record, and then searching the instances of the user on systems references in each identification entry. Thus, for every specific user, the personal assistant maintains the list of the systems the user has interacted with.

Mehta [2008] introduces a compact representation of the user's current context model, namely a *context passport*, which is exploited to identify users during their interactions with different web systems. In this way, systems are able to recognize every user and be aware of her preferences and context.

In Carmagnola and Cena [2009] the key idea is to view users as a collection of properties and to manage user identification across systems through the comparison of the values of common properties. The approach is based on a common registry (the Identification Registry) where systems subscribe themselves, specifying the user model features they exploit to profile users. As a final remark, many researchers rely on exploiting standards to unequivocally identify a user over her web interactions. An example of such a standard is OpenID [OPENID, 2007], which is an open, decentralized, free framework for verifying users' online identity, used for example by Cena and Furnari [2009].

10.3 Evaluation in the UM interoperability process

A relevant issue which has not been addressed so far concerns the evaluation of the user model enhanced with the data coming from several external sources. In fact, most of the work we have analyzed have not tackled this aspect.

The evaluation of the UM interoperability process is twofold. On the one hand, it involves establishing if the data gathered during the interoperability process are reliable. On the other hand, it involves measuring if the user model has been quantitatively and qualitatively improved. Even if evaluation is generally overlooked in the reviewed work, we have to mention that Kay et al. [2002], Kobsa and Fink [2006] carried out performance evaluations of their respective user modeling servers (Personis and UMS).

The predictive evaluation phase. As in traditional HCI approaches, predictive evaluation is aimed at *making predictions*. In a UM interoperability context, predictions may concern the reliability of the collected user model data. Such reliability can be assessed by exploiting suitable metadata about the exchanged data (such as the reasoning mechanism used to derive them, the date of the last update, and so on), and about the system providing them. For example, Carmagnola and Dimitrova [2008] suggested to exchange, together with the user model data, a set of additional metadata aimed at measuring the trustworthiness of the data. In this approach, the Inference Web Initiative[28] was designed in the context of the Semantic Web with the aim of enabling applications to generate portable and distributed justifications for any answers they produce. In a UM interoperability scenario such justifications could be used in the predictive evaluation phase.

The summative evaluation phase. As foreseen in Section 2, one assumption about UM interoperability is that it brings about a qualitative improvement of the adaptive behavior of the system [Berkovsky, 2005]. The quantitative improvement (due to the increased coverage of the model) should also be evaluated. In this perspective, summative evaluations should assess, for instance, the accuracy, the final users' opinions and satisfaction, and the coverage of the enhanced user model, respectively.

There has been little experience in evaluating such user model. We can mention the work of Berkovsky et al. [2009] and Carmagnola and Dimitrova [2008]. Both of these performed a final evaluation to determine if the user model data exchange has really improved the quality of the generated recommendations, comparing the Mean Absolute Error (MAE) before and after the exchange of user model data. Their final results showed a real improvement of the MAE due to the UM interoperability process.

11 Challenges and future trends

In this final section we sketch some open issues and future trends for UM interoperability that in our vision can inspire promising research directions and deserve future investigations: They concern the opportunities for UM interoperability related to :

- the exchange of the user modeling reasoning strategies;
- Web 2.0 scenarios;
- scrutable user models in the UM interoperability context;
- dynamic user models.

Interoperability of user modeling reasoning strategies. So far we have discussed the possibility of exchanging user model data. A promising research direction is the exchange of user modeling reasoning strategies, that is the inferences that lead to the user model data. We refer to this as *interoperable user modeling process* [Carmagnola and Cena, 2006a]. This would also support the predictive evaluation phase (see Section 10.3). In content-based adaptive systems [Brusilovsky and Millán, 2007] user modeling reasoning strategies can easily be expressed in the form of rules which have the advantage of merging adaptation methods on the basis of the user model features and of the domain model in a way that is comprehensible both to the final users and to software agents. To enable the sharing of the reasoning strategies across systems, a semantic representation of the rules and their components is required. Rule markup languages[29] have been specifically designed with the purpose of reusing rules on the Web and in other distributed systems. This could be a promising starting point for interoperable user modeling.

UM interoperability in Web 2.0 and Web 3.0. The *Web 2.0* phenomenon, coined by O'Reilly Media in 2005 [OReilly, 2005], introduces various social applications enabling

[28] http://iw.stanford.edu/2.0/

[29] Like RuleML [RuleML, 2004], SWRL [SWRL, 2004] and RIF [RIF, 2005].

online collaboration and encouraging the participation and contribution of spontaneous social networks. Web 2.0 offers different ways to easily participate in the creation of web contents: inserting new contents, sharing objects, providing comments and so on. Many social applications share the idea of tagging, which is one of the building blocks of the Web 2.0 vision. Thus, all these systems have the possibility to collect and store the tags the users used to label items. These data could be extremely useful for user-adaptive applications. Applications may exploit tags in order to enrich their user model. "Annotations can become a part of his user profile as an indication of his perspective on the content collection and interest in the annotated object" [Van Setten et al., 2006]. Tags can be semantically analyzed in order to infer knowledge about a specific user. Carmagnola et al. [2008] have proposed to map tags to domain ontologies. This would allow reasoning on tags semantics, in order to infer information (such as interests and preferences) about the user who tagged a resource, enriching her user model.

Since tags can be used to improve the user model they can be exchanged, together with the user model, among applications. Wang et al. [2008] propose to exchange user's tags across systems to improve the quality of the adaptation. More specifically, they provide a method for extracting, conceptualizing and linking user tags contained in public RSS files generated in the interaction of users with a social recommender system iCITY [Carmagnola et al., 2008]. The tags are then mapped to art-related concepts used in the personalized museum applications CHIP [Cramer et al., 2008], and then used to enrich the user profile for generating personalized recommendations of artworks and topics in CHIP. To implement the tag mapping from iCITY to CHIP, the authors exploit the Simple Knowledge Organization System (SKOS) Core Mapping Vocabulary Specification [SKOS, 2004].

However, not only tags are offered by Web 2.0 users. Comments, networks of friends, networks of interests are provided from users in many different social web sites. Indeed, it could be interesting to aggregate, interpret and exchange these other pieces of information as well.

Taking this approach, Kyriacou [2009] proposed a Scrutable User Modelling Infrastructure which enables gathering of user models constructed from daily interactions with services from the social e-networking (such as Facebook and MySpace) and e-commerce domains (such as Amazon and eBay) and exporting these models to educational personalization systems. Also the work of Abel et al. [2010] aggregate user information from heterogenous social systems to build a global user profile.

The call for interoperability and the need of interconnecting both content and people in a meaningful way is also emphasized by Braslin and Decker, who state that "by using agreed-upon semantic formats to describe people, content objects and the connections that bind them all together, social media sites can interoperate by appealing to common semantics" [Breslin and Decker, 2007].

According to Pidgin Technologies[30] social web sites have to promote profile centralization and profile interoperability. In Web 2.0 there are many isolated communities of users and their data. Moving toward Web 3.0, the so-called "intelligent web", there is a need to connect these islands, allowing users to easily move from one to another, and enabling users to easily bring their data with them and reuse their profile. This is known as *web data portability*.

Web data portability is hard to be achieved since usually social web sites represent information in their own format, and often the information is provided in natural language. Indeed, several solutions have been proposed to accomplish web data portability. One of the most

[30] http://www.pidgintech.com.

promising is Dataportability.org[31] which had the original aim of allowing users to control their own data, as it shared in different internet-based applications. More recently, the organization has achieved the further goal of advocating open standard representation of data to make web data portability a reality. There are numerous open standards that are considered to advance such a vision, such as OpenID, OAuth, microformats, FOAF, APML, RDFa, SIOC. Beside the open standards for data representation, other initiatives, such as the OpenSocial foundation[32] recently proposed by Google, Yahoo! and MySpace have provided APIs to let developers create social applications that access data from various websites.

Scrutability for UM interoperability. Currently, the scrutability of a user model involves final users who are allowed to inspect and modify their own user model [Kay, 2000]. A promising direction would be to extend the concept of system-to-user scrutability to system-to-system scrutability. In this sense, a scrutable user model may also be accessed from (allowed) applications looking for information about the user. We can imagine an interoperability scenario where trusted applications read the information contained in the *scrutable/interoperable* user model, or part of it, published on a web site. In such a vision user model data should be available in some shareable machine-readable format. In order to be defined as scrutable, the model has to be not only comprehensible, but also valuable for the machines. To this end, the various evidence proposed by Kay in PLUS [Kay, 2006] could also be used by the applications for evaluating the data in a scrutable user model.

Modeling of dynamic users. In Section 6 we have presented and discussed the kind of data that are typically exchanged during the UM interoperability process. However, for all the mentioned data, one assumption is that all external models can be considered as static during mediation. However, a user can interact with several adaptive systems at the same time and all her models continuously change. Systems need not only to locate sparse user models and to convert them, but also to map their evidence which frequently changes. Thus, the update of the user information between the systems participating in the exchange of data about the user is critical. In contrast, in modern approaches multiple user models are considered at the same time without any fusion.

The modeling of dynamic users in a range of contexts deals with the issue of “lifelong user modeling”, the ability to model a dynamic and changing user throughout lifetime interactions with a variety of resource providers [Kay, 2008]. A promising solution is to investigate how to enable the interoperability of dynamic users by merging the principles and techniques of traditional user modeling with lifelong user modeling, where UM interoperability and reuse is a key component [Kay and Kummerfeld, 2009].

Acknowledgements

We would like to thank Silvia Likavec and Rossano Schifanella for their valuable comments. We also thank all the authors who helped us in correctly describing their work and the anonymous Reviewer 3 who kindly supported us in improving the paper.

References

F. Abel, N. Henze, E. Herder, and D. Krause. Interweaving public user profiles on the web. In P. D. Bra, A. Kobsa, and D. N. Chin, editors, *User Modeling, Adaptation, and*

[31] http://www.dataportability.org/.

[32] http://www.opensocial.org/.

Personalization: 18th International Conference, UMAP 2010, volume 6075 of *Lecture Notes in Computer Science*, pages 16–27. Springer, 2010. ISBN 978-3-642-13469-2.

ACL. ACL, agent communication language. http://www.fipa.org/specs/fipa00061/index.html, 2002.

L. Ardissono, P. Brna, and A. Mitrovic, editors. *User Modeling 2005, 10th International Conference, UM 2005, Edinburgh, Scotland, UK, July 24-29, 2005, Proceedings*, volume 3538 of *Lecture Notes in Computer Science*, 2005. Springer. ISBN 3-540-27885-0.

L. Aroyo, P. Dolog, G.-J. Houben, M. Kravcik, A. Naeve, M. Nilsson, and F. Wild. Interoperability in personalized adaptive learning. *Educational Technology & Society*, 9(2): 4–18, 2006.

M. Assad, D. J. Carmichael, J. Kay, and B. Kummerfeld. PersonisAD: Distributed, active, scrutable model framework for context-aware services. In A. LaMarca, M. Langheinrich, and K. N. Truong, editors, *Pervasive*, volume 4480 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2007. ISBN 978-3-540-72036-2.

S. Berkovsky. Ubiquitous user modeling in recommender systems. In Ardissono et al. [2005], pages 496–498. ISBN 3-540-27885-0.

S. Berkovsky. Decentralized mediation of user models for a better personalization. In Wade et al. [2006], pages 404–408. ISBN 3-540-34696-1.

S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci. Privacy-enhanced collaborative filtering. In *Proceedings of the Workshop on Privacy-Enhanced Personalization (PEP), at the 10th International Conference on User Modeling (UM05)*, pages 75–83, Edinburgh, UK, 2005.

S. Berkovsky, A. Gorfinkel, T. Kuflika, and L. Manevitz. Case-based to content-based user model mediation. In *Workshop on Ubiquitous User Modeling, at the European Conference on Artificial Intelligence*, Riva del Garda, Italy, 2006a.

S. Berkovsky, T. Kuflik, and F. Ricci. Enhancing privacy while preserving the accuracy of collaborative filtering. In *Proceedings of the Workshop on Recommender Systems (RS), at the European Conference on Artificial Intelligence (ECAI)*, pages 49–53, 2006b.

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

S. Berkovsky, T. Kuflik, and F. Ricci. Cross-representation mediation of user models. *User Model. User-Adapt. Interact.*, 19(1-2):35–63, 2009.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, pages 28–37, 2001.

G. Brajnik and C. Tasso. A shell for developing nonmonotonic user modeling systems. *International Journal. Human Computer Studies*, 40:31–62, 1994.

A. Brar and J. Kay. Privacy and security in ubiquitous personalized applications. Technical report, School of Information Technologies, University of Sydney, NWS, 2004.

J. Breslin and S. Decker. The future of social networks on the internet: The need for semantics. *IEEE Internet Computing*, 11:86–90, 2007.

C. Brooks, M. Winter, J. Greer, and G. McCalla. The massive user modelling system (MUMS). In *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems*, pages 635–645, 2004.

P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Model. User-Adapt. Interact.*, 6(2-3):87–129, 1996.

P. Brusilovsky. Adaptive hypermedia. *User Model. User-Adapt. Interact.*, 11(1-2):87–110, 2001.

P. Brusilovsky. KnowledgeTree: a distributed architecture for adaptive e-learning. In S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills, editors, *WWW, International World*

Wide Web Conferences, pages 104–113. ACM, 2004. ISBN 1-58113-912-8.

P. Brusilovsky and E. Millán. User models for adaptive hypermedia and adaptive educational systems. In Brusilovsky et al. [2007], pages 3–53. ISBN 978-3-540-72078-2.

P. Brusilovsky, S. Ritter, and E. Schwarz. Distributed intelligent tutoring on the web. In *Proceedings of the 8th World Conference of the AIED Society*, pages 27–34, Kobe, Japan, August 1997.

P. Brusilovsky, A. T. Corbett, and F. de Rosis, editors. *User Modeling 2003, 9th International Conference, UM 2003, Johnstown, PA, USA, June 22-26, 2003, Proceedings*, volume 2702 of *Lecture Notes in Computer Science*, 2003. Springer. ISBN 3-540-40381-7.

P. Brusilovsky, A. Kobsa, and W. Nejdl, editors. *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, 2007. Springer. ISBN 978-3-540-72078-2.

A. K. Caglayan, M. Snorrason, J. Jacoby, J. Mazzu, R. Jones, and K. Kumar. Learn sesame, a learning agent engine. *Applied Artificial Intelligence*, 11(5):393–412, 1997.

F. Carmagnola. *From User Models to Interoperable User Models*. PhD thesis, University of Turin, Italy, December 2007.

F. Carmagnola and F. Cena. From interoperable user models to interoperable user modeling. In Wade et al. [2006], pages 409–413. ISBN 3-540-34696-1.

F. Carmagnola and F. Cena. An approach for evaluating user model data in an interoperability scenario. In *Proceedings of European Starting AI Researcher Symposium, STAIRS*, pages 251–252, Riva del Garda (TN), Italy, August 28-29 2006b. IOS PRESS. ISBN 1-58603-645-9.

F. Carmagnola and F. Cena. User identification for cross-system personalisation. *Information Sciences*, 179(1-2):16–32, 2009.

F. Carmagnola and V. Dimitrova. An evidence-based approach to handle semantic heterogeneity in interoperable distributed user models. In Nejdl et al. [2008], pages 73–82. ISBN 978-3-540-70984-8.

F. Carmagnola, F. Cena, L. Console, O. Cortassa, C. Gena, A. Goy, I. Torre, A. Toso, and F. Vernero. Tag-based user modeling for social multi-device adaptive guides. *User Model. User-Adapt. Interact.*, 18(5):497–538, 2008.

F. Cena. *The role of semantic dialogue for adaptation in ubiquitous environment*. PhD thesis, Department of Computer Science, Torino, Italy, March 2007.

F. Cena and R. Furnari. A model for feature-based user model interoperability on the web. In Kuflik et al. [2009], pages 37–54. ISBN 978-3-642-05038-1.

V. I. Chepegin, L. Aroyo, and P. D. Bra. Discovery service for user models in a multi-application context. In *ICALT*, pages 956–957. IEEE Computer Society, 2005. ISBN 0-7695-2338-2.

C. Conati, K. F. McCoy, and G. Paliouras, editors. *User Modeling 2007, 11th International Conference, UM 2007, Corfu, Greece, June 25-29, 2007, Proceedings*, volume 4511 of *Lecture Notes in Computer Science*, 2007. Springer. ISBN 978-3-540-73077-4.

H. S. M. Cramer, V. Evers, S. Ramlal, M. van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. J. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Model. User-Adapt. Interact.*, 18(5):455–496, 2008.

L. F. Cranor. 'I didn't buy it for myself' privacy and ecommerce personalization. In *WPES '03: Proceedings of the 2003 ACM workshop on Privacy in the electronic society*, pages 111–117, New York, NY, USA, 2003. ACM. ISBN 1-58113-776-1.

A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12:303–319, November 2003. ISSN

1066-8888.

P. Dolog. Identifying relevant fragments of learner profile on the semantic web. In *Workshop on Applications of Semantic Web Technologies for E-learning - SW-EL'04 at ISWC 2004*, Hiroshima, Japan, 2004.

P. Dolog and M. Schäfer. A framework for browsing, manipulating and maintaining interoperable learner profiles. In Ardissono et al. [2005], pages 397–401. ISBN 3-540-27885-0.

P. Dolog and M. Schäfer. Learner modeling on the semantic web. In *PerSWeb-2005 Workshop: Personalization on the Semantic Web 2005: 10th International Conference, UM05*, Edinburgh, Scotland, 2005b.

P. Dolog and J. Vassileva. Decentralized, agent based and social approaches to user modelling. In *Workshop DASUM-05, at the 9th International Conference on User Modeling (UM'05)*, Edinburgh, Scotland, August 2005.

W. Edwards. Discovery systems in ubiquitous computing. *Pervasive Computing, IEEE*, 5: 70–77, 2006.

J. Fink. *User Modeling Servers - Requirements, Design and Evaluation*. PhD thesis, University of Duisburg-Essen, 2003.

J. Fink and A. Kobsa. A review and analysis of commercial user modeling servers for personalization on the World Wide Web. *User Model and User-Adapted Interaction*, 10 (2-3):209–249, 2000.

FIPA. Directory Facilitator and Service Discovery Agent. http://www.fipa.org/docs/input/f-in-00070/f-in-00070.pdf, 2002.

E. Guttman. Discovery systems in ubiquitous computing. *IEEE Pervasive Computing*, 5(2): 70–77, 2006.

D. Heckmann. *Ubiquitous User Modeling*. PhD thesis, Department of Computer Science Saarbrucken, Saarland University, 2005.

D. Heckmann and A. Krüger. A user modeling markup language (UserML) for ubiquitous computing. In Brusilovsky et al. [2003], pages 393–397. ISBN 3-540-40381-7.

D. Heckmann, T. Schwartz, B. Brandherm, and A. Kröner. Decentralized user modeling with UserML and GUMO. In P. Dolog and J. Vassileva, editors, *Proceedings of the Workshop on Decentralized, Agent Based and Social Approaches to User Modeling, DASUM-05, at UM2005*, pages 61–66, Edinburgh, Scotland, Jul 2005a.

D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. Gumo - the general user model ontology. In Ardissono et al. [2005], pages 428–432. ISBN 3-540-27885-0.

T. Howes, M. Smith, and G. Good. *Understanding and Deploying LDAP Directory Services*. Indianapolis, USA, 1999.

J. Kadane and D. Schum. *A Probabilistic Analysis of the Sacco and Vanzetti Evidence*. J. Wiley and Sons, 1996. ISBN 0471141828.

J. Kay. The um toolkit for reusable, long term user models. In *User Modelling and User-adapted Interaction*, volume 4, pages 149–196, 1995.

J. Kay. Accretion representation for scrutable student modelling. *Intelligent Tutoring Systems*, pages 514–523, 2000.

J. Kay. Scrutable adaptation: Because we can and must. In Wade et al. [2006], pages 11–19. ISBN 3-540-34696-1.

J. Kay. Lifelong learner modeling for lifelong personalized pervasive learning. *IEEE Transaction on Learning Technologies*, 1(4):215–228, 2008.

J. Kay and B. Kummerfeld. Scrutability, user control and privacy for distributed personalization. In *Proceedings of Workshop on Privacy-Enhanced Personalization, at the International Conference on Human Factors in Computing Systems*, Québec, Canada, April

2006.

J. Kay and B. Kummerfeld. Lifelong user modelling goals, issues and challenges. In *Proceedings of the Lifelong User Modelling Workshop, at UMAP'09 User Modeling Adaptation, and Personalization*, Trento, Italy, 2009.

J. Kay, B. Kummerfeld, and P. Lauder. Personis: A server for user models. In P. D. Bra, P. Brusilovsky, and R. Conejo, editors, *AH*, volume 2347 of *Lecture Notes in Computer Science*, pages 203–212. Springer, 2002. ISBN 3-540-43737-1.

A. Kobsa. Generic user modeling systems. *UMUAI, User Modeling and User-Adapted Interaction*, 11:49–63, 2001.

A. Kobsa. Privacy-enhanced web personalization. In Brusilovsky et al. [2007], pages 628–670. ISBN 978-3-540-72078-2.

A. Kobsa. Generic user modeling systems. In Brusilovsky et al. [2007], pages 136–154. ISBN 978-3-540-72078-2.

A. Kobsa and J. Fink. An ldap-based user modeling server and its evaluation. *User Modelling and User-Adaptive Interaction*, 16(2):129–169, 2006.

A. Kobsa and W. Pohl. The user modeling shell system BGP-MS. *User Model. User-Adapt. Interact.*, 4(2):59–106, 1995.

A. Kobsa and J. Schreck. Privacy through pseudonymity in user-adaptive systems. *ACM Transactions on Internet Technology*, 3(2):149–183, 2003.

A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving online customer relationship. *The Knowledge Engineering Review*, 16(2): 111–155, 2001.

J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to Usenet news. *Commun. ACM*, 40(3):77–87, 1997.

T. Kuflik, S. Berkovsky, F. Carmagnola, D. Heckmann, and A. Krüger, editors. *Advances in Ubiquitous User Modelling, Revised Selected Papers*, volume 5830 of *Lecture Notes in Computer Science*. Springer, 2009. ISBN 978-3-642-05038-1.

D. Kyriacou. Enriching lifelong user modelling with the social e-networking and e-commerce "pieces of the puzzle". In *First and Seventeenth International Conference on User Modeling, Adaptation, and Personalization (UMAP2009) - Doctoral Consortium*, pages 428–434, Trento, Italy, 2009.

R. LaRose, N. J. Rifon, and A. Lee. Promoting I-safety: Effects of privacy warning boxes and privacy seals on risk assessment and online privacy behaviors. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189, Boston, Massachusetts, June 2004.

LIP. Lip, learner information package, version 1.0.1. http://www.imsglobal.org/profiles/index.html, 2005.

A. Lorenz. Agent-based ubiquitous user modeling. In Ardissono et al. [2005], pages 512–514. ISBN 3-540-27885-0.

I. Machado, A. Martins, and A. Paiva. One for all and all in one: a learner modelling server in a multi-agent platform. In *UM '99: Proceedings of the seventh international conference on User modeling*, pages 211–221, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc. ISBN 3-211-83151-7.

D. Martin, M. Burstein, D. Mcdermott, S. Mcilraith, M. Paolucci, K. Sycara, D. L. Mcguinness, E. Sirin, and N. Srinivasan. Bringing semantics to web services with OWL-S. *World Wide Web*, 10:243–277, September 2007. ISSN 1386-145X.

G. McCalla, J. Vassileva, J. Greer, and S. Bull. Active learner modelling. *Intelligent Tutoring Systems*, pages 53–62, 2000.

B. Mehta. Learning from what others know: Privacy preserving cross system personalization. In Conati et al. [2007], pages 57–66. ISBN 978-3-540-73077-4.

B. Mehta. *Cross system personalization: Enabling personalization across multiple systems*. PhD thesis, Department of Computer Science, Univesity of Duisberg, Germany, February 2008.

B. Mehta, C. Niederée, A. Stewart, M. Degemmis, P. Lops, and G. Semeraro. Ontologically-enriched unified user modeling for cross-system personalization. In Ardissono et al. [2005], pages 119–123. ISBN 3-540-27885-0.

D. L. Musa and J. P. M. de Oliveira. Sharing learner information through a web services-based learning architecture. *Journal of Web Engeneering*, 4(3):263–278, 2005.

W. Nejdl, J. Kay, P. Pu, and E. Herder, editors. *Adaptive Hypermedia and Adaptive Web-Based Systems, 5th International Conference, AH 2008, Hannover, Germany, July 29 - August 1, 2008. Proceedings*, volume 5149 of *Lecture Notes in Computer Science*, 2008. Springer. ISBN 978-3-540-70984-8.

X. Niu, G. I. McCalla, and J. Vassileva. Purpose-based user modelling in a multi-agent portfolio management system. In Brusilovsky et al. [2003], pages 398–402. ISBN 3-540-40381-7.

X. Niu, G. I. McCalla, and J. Vassileva. Purpose-based expert finding in a portfolio management system. *Computational Intelligence Journal*, 20(4):548–561, 2004.

OPENID. Openid initiative. http://openid.net/foundation/, 2007.

T. OReilly. What is web 2.0. design patterns and business models for the next generation of software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, 2005.

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

A. Paiva and J. A. Self. Tagus - a user and leamer modeling workbench. *User Model. User-Adapt. Interact.*, 4(3):197–226, 1995.

PAPI. Papi specification, public and private information. http://ltsc.ieee.org/wg2, 2003.

S. Ram and J. Park. Semantic conflict resolution ontology (scrol): An ontology for detecting and resolving data and schema-level semantic conflicts. *IEEE Trans. Knowl. Data Eng.*, 16(2):189–202, 2004.

RIF. Rif working group. http://www.w3.org/2005/rules/, 2005.

RuleML. The rule markup initiative. http://www.ruleml.org/, 2004.

J. Schreck. *Security and Privacy in User Modeling*. PhD thesis, Dordrecht, Netherlands, http://www.security-and-privacy-in-user-modeling.info, 2003.

E. Sheikh and J. Sticklen. Generating intelligent tutoring systems from reusable components and knowledge-based systems. In *ITS '02: Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, pages 199–207, London, UK, 2002. Springer-Verlag. ISBN 3-540-43750-9.

SKOS. SKOS - Simple Knowledge Organization System. http://www.w3.org/2004/02/skos/, 2004.

S. A. Sosnovsky, P. Brusilovsky, M. Yudelson, A. Mitrovic, M. Mathews, and A. N. Kumar. Semantic integration of adaptive educational systems. In Kuflik et al. [2009], pages 134–158. ISBN 978-3-642-05038-1.

S. Staab and H. Stuckenschmidt. *Semantic Web and Peer-to-Peer Decentralized Management and Exchange of Knowledge and Information*. Springer Verlag, Berlin, 2006. ISBN 978-3-540-28346-1.

C. Stewart, A. Cristea, I. Celik, and E. Ashman. Interoperability between aeh user models. In *Proceedings of International workshop on Adaptivity, personalization and the semantic* *web, in Conference on Hypertext and Hypermedia*, pages 21–30, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-453-7.

SWRL. SWRL: A semantic web rule language combining owl and ruleml. http://www.w3.org/Submission/SWRL/, 2004.

M. Trella, R. Conejo, E. Guzmán, and D. Bueno. An educational component based framework for web its development. In J. M. C. Lovelle, B. M. G. Rodríguez, L. J. Aguilar, J. E. L. Gayo, and M. del Puerto Paule Ruíz, editors, *ICWE*, volume 2722 of *Lecture Notes in Computer Science*, pages 134–143. Springer, 2003. ISBN 3-540-40522-4.

UDDI. UDDI Spec TC. http://uddi.org/pubs/uddi-v3.0.2-20041019.htm, 2004.

K. van der Sluijs and G.-J. Houben. A generic component for exchanging user models between web-based systems. *International Journal of Continuing Engineering Education and Life-Long Learning (IJCEELL)*, 16(1-2):64–76, 2006.

M. Van Setten, R. Brussee, H. van Vliet, L. Gazendam, Y. van Houten, and M. Veenstra. On the importance of "who tagged what". In *Proceedings of the Workshop on Social Navigation and Community based Adaptation Technologies at AH 2006, the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 552–561, Dublin, Ireland, 2006.

J. Vassileva. Distributed user modelling for universal information access. In C. Stephanidis, editor, *Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction*, pages 122–126. Lawrence Erlbaum, 2001. ISBN 0-8058-3609-8.

W3C. Wsdl - web service description language. http://www.w3.org/TR/wsdl, 2001.

V. P. Wade, H. Ashman, and B. Smyth, editors. *Adaptive Hypermedia and Adaptive Web-Based Systems, 4th International Conference, AH 2006, Dublin, Ireland, June 21-23, 2006, Proceedings*, volume 4018 of *Lecture Notes in Computer Science*, 2006. Springer. ISBN 3-540-34696-1.

W. Wahlster and A. Kobsa. User models in dialog systems. In *User Models in Dialog Systems*, pages 4–34. Springer-Verlag, 1989.

Y. Wang and A. Kobsa. Respecting users' individual privacy constraints in web personalization. In Conati et al. [2007], pages 157–166. ISBN 978-3-540-73077-4.

Y. Wang, F. Cena, F. Carmagnola, O. Cortassa, C. Gena, N. Stash, and L. Aroyo. Rss-based interoperability for user adaptive systems. In Nejdl et al. [2008], pages 353–356. ISBN 978-3-540-70984-8.

P. Wegner. Interoperability. *Computing Surveys (CSUR)*, 28:285–287, 1996.

M. Weiser. The future of ubiquitous computing on campus. *Communications of the ACM*, 41(1):41–42, 1998.

W. Wörndl and M. Koch. Privacy in distributed user profile management. In *WWW (Posters)*, http://www2003.org/cdrom/papers/poster/p020/p20-woerndl.htm, 2003.

D. Yimam-Seid and A. Kobsa. Expert finding systems for organizations: Problem and domain analysis andthe demoir approach. *Journal of Organizational Computing and Electronic Commerce*, 13(1):1–24, 2003.

M. Yudelson, P. Brusilovsky, and V. Zadorozhny. A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In Conati et al. [2007], pages 27–36. ISBN 978-3-540-73077-4.

F. Zhang, Z. Song, and H. Zhang. Web service based architecture and ontology based user model for cross-system personalization. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 849–852, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0769527477.

Author Biographies

Francesca Carmagnola is a Researcher in Computer Science at the University of Torino She received her PhD in 2007 in Communication Science (University of Torino), in the area of distributed and interoperable user models. Her current research activities address the user modeling, personalization of adaptive hypermedia systems, personalization of ubiquitous systems, infrastructures for adaptive web-based systems, personalized digital television, semantic knowledge representation and reasoning, semantic web.

Federica Cena is a Researcher at the Department of Computer Science of the University of Torino. She achieved her PhD at University of Torino in Communication Science in 2007, in the area of semantic web for distributed web applications. She is working on user modeling, personalization, ubiquitous computing applications and web services. In the last years, she is mainly devoted in studying the implications of Social Web and Semantic Web for user modeling.

Cristina Gena is Researcher at the Department of Computer Science of the University of Torino, working in the area of intelligent user interfaces. She completed her Ph.D. in Communication Science (University of Torino) in 2003, with a thesis on the evaluation of user-adaptive systems.Her current research activities address user modeling, adaptive web systems and their evaluation, context-aware systems, semantic web, web 2.0, usability and interaction design. Her contribution is based on experiences gained within these fields.