

The FUSE Domain-Aware Approach to User Model Interoperability: A Comparative Study

Eddie Walsh, Alexander O'Connor and Vincent Wade
Knowledge and Data Engineering Group
School of Computer Science and Statistics
Trinity College Dublin, Ireland
{eddie.walsh,alex.oconnor,vincent.wade}@scss.tcd.ie

Abstract

Exchanging user information between multiple sources can potentially bring many benefits that enhance the user experience in software applications. Richer and more dynamic user models can be constructed allowing more tailoring of content and services to the specific needs of individual users. However, providing effective interoperability is a complex challenge as there can be significant heterogeneity between user models. Current generic approaches and tools offer insufficient support for the complex domain-specific nature of many user model interoperability tasks. This paper presents a comparative study of the FUSE domain-aware approach to user model interoperability. This novel approach differs from existing generic approaches by incorporating domain knowledge in new processes and tools to support complex user model interoperability tasks in multiple overlapping domains. The FUSE approach is described and compared to existing state of the art approaches using a consistent set of key issues in user model interoperability.

1 Introduction

Substantial amounts of information are now being gathered about users [8]. Exchanging this user information between multiple sources can potentially provide many benefits to enhance the end-user experience [3]. Richer and more dynamic user models can be constructed and shared across multiple systems. This allows each system to provide improved user-centric functionality while reducing duplicate user interactions [5]. One of the key benefits of user model interoperability is in the provision of personalisation. The sharing of richer user models allows for greater tailoring of content and services to each individual user's specific needs and preferences [5].

However, there are many organisational and technologi-

cal boundaries that inhibit the sharing and reuse of this information [8]. User models are designed for different purposes and, as a result, there can be significant heterogeneity at the structural, syntactic and semantic levels. Standardised user models, for example IMS LIP in education, have been developed but heterogeneity remains a major challenge [19]. An alternative method to standardisation involves using mediation to overcome user model heterogeneity [8]. This can consist of a manual mapping process where an administrator identifies relationships between equivalent elements in different data models. This process can be supported using one of the many generic mapping tools available, such as Altova Mapforce [1]. Matching tools can also be used to help find potential matches between user models automatically [18]. However, at present the complexity of many existing data interoperability tasks means that matching cannot complete the entire mediation process automatically [12]. As a result, manual mapping provided by an administrator continues to be a significant requirement for user model interoperability.

To provide more expansive and detailed user models requires the exchange of user information from multiple overlapping domains. In this research, a domain is defined as a group of related terms that represent a particular field of knowledge. For example, a user model can contain general user information from the user domain (e.g identification, interests, preferences). This can overlap with user information from other domains such as education (e.g. learning preferences, assessments) or employment (e.g performance, responsibilities). Subsequently, these domains can also overlap with related user information from particular subject domains (e.g competencies in mathematics, assessments in computer literacy). This overlap of domains within user models increases the complexity of the manual mapping task: requiring mapping administrators that possess increasingly specialised domain knowledge. Current generic mapping approaches and tools offer insufficient support for

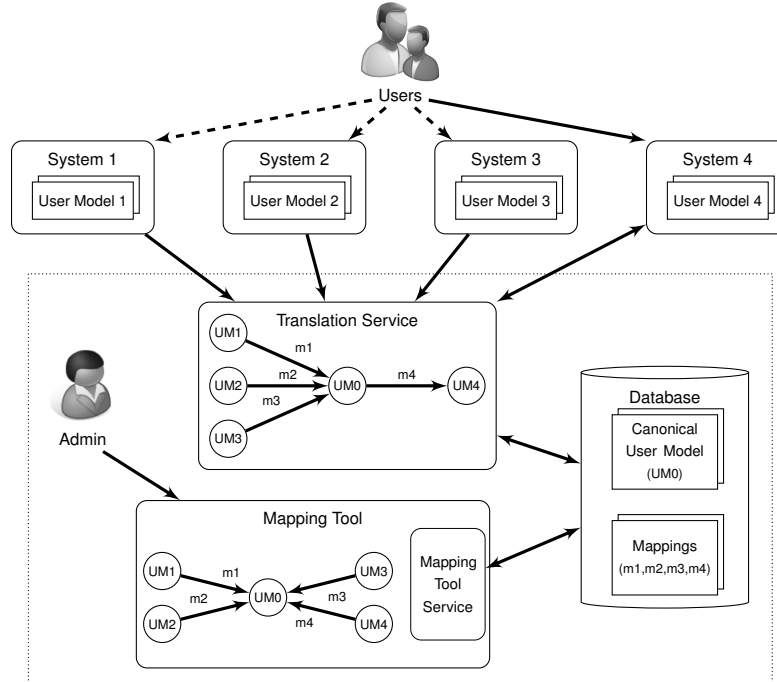


Figure 1: FUSE architecture

the domain-specific nature of many user model interoperability tasks.

In this paper, the FUSE domain-aware approach [20] to user model interoperability is described. A domain-aware approach can be defined as the incorporation of exchangeable knowledge from one or more domains into generic tools and processes to support complex interoperability tasks. The FUSE approach uses knowledge from multiple domains to support both manual mapping and automatic translation for user model interoperability. Following this, a comparative study is presented to contrast FUSE with a selection of existing state of the art approaches. This study is conducted across a consistent set of key issues to provide a systematic evaluation of FUSE in the context of the related research in user model interoperability.

2 FUSE: A Domain-Aware Approach to User Model Interoperability

The FUSE (Federated User Exchange) approach [20] consists of two processes: a manual mapping process and an automatic translation process. Both processes contain two domain-aware mechanisms: (i) a canonical user model and (ii) user model mapping transforms, which tailor the processes to specific domains. All mappings are created to the canonical user model which is a consistent shared user model representation. The user model mapping transforms are mapping components specifically created and used for

mapping between user models. An architecture which applies the FUSE domain-aware approach in both the mapping and translation processes is shown in Figure 1.

2.1 Mapping Tool

As shown in Figure 1, the Mapping Tool allows the creation of relationships between different types of user models. User model schemas and sample instances can be uploaded to the Mapping Tool (UM1, UM2, UM3, UM4). Mappings from elements in the user models to their equivalent elements in the canonical user model (UM0) are defined manually using the graphical user interface, also shown in Figure 2. The Mapping Tool can then automatically generate executable versions of the mappings (m1, m2, m3, m4). The Mapping Tool stores the various schemas, instances and mappings in the Database. The Mapping Tool communicates with the Database using the Mapping Tool Service. This service allows various tasks to be completed such as saving and retrieving user models and executing mappings. The Mapping Tool Service along with the Database are server-side components whereas the Mapping Tool is a client-side component. Once the mappings have been completed in the Mapping Tool and deployed to the Database, they are available to the Translation Service for use during the automatic user model translation process.

The Mapping Tool supports the domain-aware approach through the provision of the two domain-specific mecha-

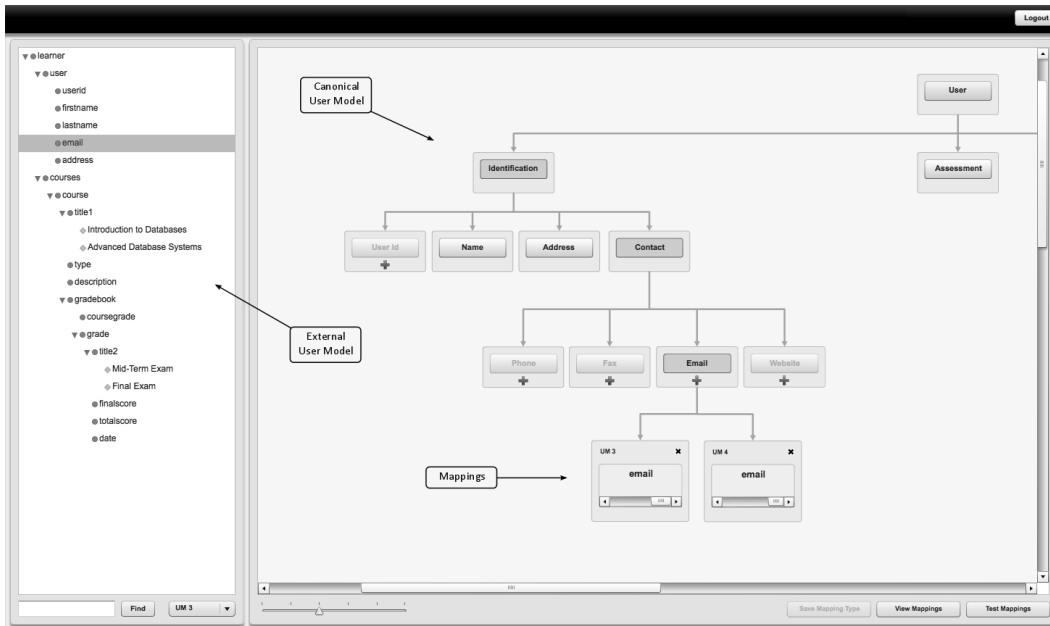


Figure 2: Mapping Tool

nisms: the canonical user model and user model mapping transforms. In the Mapping Tool, the canonical model is used in the visual creation of the mappings. This is fundamentally different to existing mapping tools as it allows a clearer and more consistent visualisation, allowing the administrator to continually map to the same user model structure. The user model mapping transforms are also used in the visual creation of mappings. This approach allows the administrator to generate and reuse specialised user model mappings to support domain-specific mapping tasks.

Table 1 presents a representative sample of the type of user model mapping transforms that can be created in the overlapping domains of user, education, databases and SQL. These user model mapping transforms are formed using various generic schema, instance and functional mapping transforms. The generic mapping transforms are combined with domain-specific knowledge to provide user model mapping transforms that are specifically designed for heterogeneous user models in particular application areas. These user model mapping transforms can support an administrator in typically complex user model interoperability tasks across multiple overlapping domains. The user model mapping transforms can be reused by the administrator or shared with other administrators to build a large collection of relevant mappings in an application area of interest.

2.2 Translation Service

When the administrator-led user model mapping process has been completed for a number of systems, the user model

translation process can be performed to automatically exchange user information between those systems.

As shown in Figure 1, the Translation Service is a tool to automatically translate user model instances between external systems (UM1, UM2, UM3, UM4). The Translation Service component has access to the mappings that were created using the Mapping Tool and stored in the Database (m1, m2, m3, m4). These mappings represent executable relationships between the external user model elements and their equivalent elements in the canonical user model (UM0). The Translation Service automatically retrieves instances of the various external user models and temporarily stores them in the Database. It then executes the mappings to transform user information between the heterogeneous user model representations. This updates the required target user model instance which is then returned to the requesting system. The Translation Service communicates with the Database directly as they are both server-side components.

The Translation Service supports the domain-aware approach through the provision of the two domain-specific mechanisms: the canonical user model and the user model mapping transforms. The canonical user model approach adds an additional step to the translation process but ultimately reduces the number of mappings required in multi-system interoperability. The user model mapping transforms are created to map user model information in specific domains. Their use can reduce the overall number of mapping transforms required in the translation process.

Table 1: Examples of user model mapping transforms

No.	User Model Mapping Transform	Example
1	equivalent first name (A1 schema equivalence)	firstname = forename
2	join first name and last name (A2 schema join)	firstname + lastname = fullname
3	split address and get country (A3 schema split)	address[street,city,country] = country
4	equivalent learning style (B1 instance equivalence)	type[reflector]/score[10] = value[10]
5	join SQL concepts (B2 instance join)	concept[SQL1] + concept[SQL2] = concept[SQLA]
6	split SQL concepts (B3 instance split)	concept[SQL1] = concept[SQLA] + concept[SQLB]
7	multiple equivalent user ids (B4 instance multiple)	userid[jsmith, rjones] = userid[06125,00242]
8	assessment date format (C1 functional format)	date[31/01/2012] = date[01-31-2012]
9	grades 0-20 to 0-100 (C2 functional numeric)	score[18] * 5 = result[90]
10	grades 0-20 to Pass/Fail (C3 functional interval)	score[11-20,0-10] = grade[Pass,Fail]
11	SQL concept to 0/1 score (C4 functional detection)	concept[] = score[0] or concept[SQL1] = score[1]

3 Comparative Study of FUSE and Related Approaches

To evaluate the overall effectiveness of the FUSE domain-aware approach, it has been compared with a selection of existing state of the art approaches. The comparison is conducted using six key issues for user model interoperability identified by Carmagnola et al. [8]. The methods used in the FUSE approach to address the key issues are discussed in the following sections in relation to the surveyed approaches. To summarise the evaluation, Table 2 presents each key issue and the methods used by each approach.

1. **Interoperability task** describes the intended purpose of the interoperability system and the main function that will be performed. Generally, approaches have either facilitated the exchange of user models or provided a centralised user modelling/adaptation service that stores and maintains user information for individual systems.

The selection of an appropriate method depends mainly on the intended purpose of the interoperability system. The interoperability task of the FUSE approach is to exchange user models. The majority of other surveyed approaches also perform this function. The approaches that provide a user modelling/adaptation service are Personis, CUMULATE and GUMF. These approaches are designed to provide centralised management of user modelling across various systems.

2. **Interoperability architecture** specifies the overall architectural design that is used to exchange the user models. This key issue is closely related to the interoperability task and the main designs are the centralised, decentralised and mixed architectures.

There are advantages and disadvantages associated with each architectural design. The centralised approaches such as Personis, CUMULATE and GUMF, provide common representations and locations which make it easier to manage multiple user models. However, the centralised management inhibits flexibility and wider applicability in the exchange of user models.

Decentralised approaches such as the UUCM and GUMO/UserML provide more flexible interoperability where different combinations of systems can exchange user models directly. However, they rely on the standardisation of the user models to support exchange which inhibits more widespread use. Other decentralised approaches such as Carmagnola and Dimitrova's approach and Cena's approach attempt to provide automatic translation of user information. However, these approaches rely on the user data in different systems being similar and highly structured to allow matches to be determined semi or fully automatically. These approaches are not easily applicable to existing systems and user models which are likely to contain high levels of data heterogeneity.

The mixed architecture is adopted in the FUSE approach

Table 2: Comparison of FUSE with surveyed approaches

Approach	Interoperability task	Interoperability architecture	Protocols and languages for communication	Representation of data exchange	Exchanged data type	Integration of exchanged data
FUSE [20]	Exchange user models	Mixed	REST XML	Common user model and translation	User, usage, environment, domain, inferred, social	Merging of data
Personis [16] [4]	User modelling service	Centralised	HTTP JSON	Common user model	User, environment, inferred	No merge of data
CUMULATE [7] [21]	User modelling service	Centralised	HTTP XML	Common user model and translation	User, usage, domain, inferred	No merge of data
GUMF [2]	User modelling service	Centralised	REST/SOAP RDF	Common user model and translation	User, usage, environment, domain, inferred	No merge of data
UUCM [17]	Exchange user models	Decentralised	HTTP CSCP	Common user model	User, usage, environment, social	No merge of data
GUMO/UserML [14] [15]	Exchange user models	Decentralised	HTTP UserML, OWL	Common user model	User, usage, environment, inferred	No merge of data
Carmagnola and Dimitrova [9]	Exchange user models	Decentralised	HTTP RDF, RDFS	Translation	User, usage, domain, inferred, reasoning	Merging of data
Cena [10]	Exchange user models	Decentralised	SOAP RDF, RDFS	Common user model and translation	User, domain	No merge of data
Dolog and Schafer [11]	Exchange user models	Centralised or Mixed	SOAP RDF, RDFS	Common user model	User, usage, environment, domain, inferred	Merging of data
GUC [19]	Exchange user models	Mixed	HTTP OWL	Common user model and translation	User, usage, environment, inferred	No merge of data
Berkovsky [6]	Exchange user models	Mixed	n/a	Common user model and translation	User, usage, environment, inferred, social	No merge of data

and also by GUC and Berkovsky's approach. Mixed approaches provide more flexibility and applicability to existing user models by providing translation between different representations. A common centralised component reduces the complexity in the management of the user model translation. The common user model and translation approaches are discussed in more detail in the related key issue: the representation of data exchange.

3. **Protocols and languages for communication** describes the methods used to exchange the user models between systems.

HTTP and web services are the most commonly employed methods by the surveyed approaches to exchange the user models. The most common representation format for user models amongst the surveyed approaches is RDF. Some other XML-based languages have been developed which provide specialised user model structure and syntax such as UserML.

In the FUSE approach, RESTful web services and HTTP are the main basis for the exchange of user models. The emergence of common web-based exchange methods across the majority of approaches is a positive aspect for the wider provision of user model interoperability. The RESTful web service method is suitable for the exchange of XML user models and is widely supported in many web-based systems.

For the user model representation, the FUSE approach uses XML. This contrasts with some of the surveyed approaches which use RDF. FUSE was designed to support existing systems with heterogeneous forms of user information. RDF is currently not supported in many systems which were key targets for interoperability. For example in education, learning management systems have extensive support for web services and XML but little or no support for RDF. As a result, an approach based on XML was selected to provide more widespread applicability to existing systems and different types of user information.

4. **Representation of data exchange** describes how the user models are represented during the exchange process from a syntactic and semantic perspective. It describes the two main methods used to represent the exchanged user data; the common user model or translation approach.

Approaches such as Personis, GUMO/UserML, UUCM, and Dolog and Schafer's approach rely on a common representation of the user model and do not provide translations between representations. While this approach promotes standardisation of user models, it does not support existing heterogeneous user models.

Approaches such as CUMULATE, GUMF, GUC, Carmagnola and Dimitrova's approach, Cena's approach,

and Berkovsky's approach provide translation between different user model representations. They also typically incorporate a common user model that provides a shared user model representation across systems. To specify the translations, many of the approaches require mappings between user models. These can be defined either by manual mapping or by employing one of the current generic mapping tools available such as Altova Mapforce [1]. These tools are often generic and can be difficult to use for this complex and specialised task [13].

The representation of the data exchange in the FUSE approach is in the form of a canonical user model combined with a translation approach. A mappings-based approach was selected to provide the translation process. These methods provide a flexible approach that can be applied to different existing systems and user model representations. This key issue is also where the FUSE approach provides a significant difference compared to the surveyed approaches.

The FUSE approach provides support in the complex domain-specific tasks inherent in user model interoperability. This is provided through the incorporation of the two domain-aware mechanisms: the canonical user model and the user model mapping transforms. These mechanisms provide domain knowledge that can support the complexity of providing user model interoperability. The FUSE Mapping Tool and Translation Service have been specifically designed and developed to incorporate these domain-aware mechanisms. This contrasts with many of the surveyed approaches which do not provide custom tools or techniques to support complex domain-specific user model interoperability tasks.

5. **Exchanged data type** specifies the types of data that can be exchanged by the interoperability system. Examples of data types include user, usage, environment, domain, inferred, reasoning and social data [8].

The exchanged data types are largely dependent on the key issue: the representation of data exchange. As the surveyed approaches use various methods for the representation of data exchange they vary in the data types that they support. For example, UUCM and GUMO/UserML attempt to provide standardised representations of the user models. These approaches restrict the supported data types to those which have been explicitly defined in the standardised model. These data types are then difficult to modify to support different data types. Translation approaches with common models, such as CUMULATE and Berkovsky's approach generally support a wider range of data types. The information is translated and a shared user model representation is used which can be expanded or modified as needed to support different data types. Mapping-based

translation approaches, such as GUMF and GUC have the potential to support the majority of the data types because new mappings can be created to support new data forms. These types of approaches can be restricted by the associated common user model representations.

In the case of the FUSE approach, a common canonical model and mapping-based translation method was selected. As a result, this provides flexibility in the support of the majority of different types of user data. The canonical model can also be extended as needed to support different forms of data. The FUSE approach has been primarily applied using forms of user and domain-specific data in the application area of education.

6. **Integration of exchanged data** describes the incorporation of the exchanged data into the systems. In general, there are two high level approaches; the exchanged user data is not merged into the systems' existing user data but is retrieved as needed and the exchanged user data is merged into existing user data.

In the majority of the related approaches there is no merging provided into existing user data. Many of the approaches do not consider this as being relevant to the core exchange process. They view it as the responsibility of the interoperable system to incorporate the exchanged data as needed [8]. Two approaches which do support the integration of exchanged data are Dolog and Schafer's approach and Carmagnola and Dimitrova's approach which attempt to resolve conflicts and merge exchanged user data.

The FUSE approach provides merging of exchanged user data into the interoperable systems. This functionality is a result of the mapping-based approach and selected technologies. The mappings are manually specified, thereby allowing an administrator to identify and resolve potential conflicts in exchanged user data. The administrator can also simulate the potential results of mappings by testing with sample user model instances. The selection of XML and XQuery for user model and mapping representations also allows update mechanisms to be used which merge exchanged user data into existing user models. Providing integration of exchanged data is a key benefit when providing interoperability between existing heterogeneous systems.

4 Discussion

The study was concerned with the comparison of the overall FUSE approach, consisting of both the Translation Service and Mapping Tool, with the current state of the art approaches for user model interoperability.

As shown in the comparison, the FUSE approach provides a high standard method for each key issue addressed.

The rationale for the selected methods was based on the findings of an extensive state of the art review. FUSE provides a mapping-based approach which uses a canonical user model as a common representation. This mixed architecture method combines the flexibility and complex interoperability provided by the mappings with the structure and consistency of the shared user model. The selection of a mapping-based approach allows the mapping of the majority of user data types. The selection of RESTful web services and XML as the communication protocols provides interoperability between many existing systems. The selection of XQuery and XML as the mapping and user model representation languages also supports the merging of data into existing XML-based user models. This results in a comprehensive overall approach which compared to the surveyed approaches provides high standard methods across six of the key issues of user model interoperability.

The FUSE approach provides its most significant contributions in key issue 4: the representation of data exchange. In this case, FUSE provides a domain-aware approach consisting of custom tools for both user model mapping and translation. Both tools support the two domain-aware mechanisms: the canonical user model and user model mapping transforms. These mechanisms provide support for the complex domain-specific nature of user model interoperability tasks. Previous experiment results have indicated that this approach can improve accuracy and usability while maintaining duration times in specific user model mapping tasks [20]. This comparison was performed against a generic mapping approach applied using a commercial mapping tool, Altova Mapforce. Mapforce is representative of the type of tool used to provide mappings in existing approaches to user model interoperability. Experiment results have also indicated that the FUSE approach can provide efficient performance in user model translation.

Overall, FUSE provides a comprehensive approach across the selected key issues for user model interoperability when compared to existing state of art approaches. It also provides indications of significant improvements in one of the key issues: the representation of data exchange.

5 Conclusions

Providing effective user model interoperability is a major challenge as there can be significant heterogeneity between user models in multiple overlapping domains. Current generic approaches and tools offer insufficient support for the complex domain-specific nature of many user model interoperability tasks. This paper has presented a systematic comparison of the FUSE domain-aware approach against the current state of the art approaches to user model interoperability. Through the analysis of the key issues and how they have been applied in the state of the art approaches,

the FUSE approach was categorised and assessed using the same criteria. The results have shown that the FUSE approach provides comprehensive user model interoperability in six key issues through the combination of the Mapping Tool and Translation Service. Previous results have also indicated that the FUSE domain-aware approach and tools provide improvements in accuracy and usability when compared with an existing approach and tool while also maintaining efficient durations and performance. This suggests that more specialised processes and tools can deliver improvements and are required to support the more widespread provision of user model interoperability.

Acknowledgments.

This work was funded by Science Foundation Ireland via grant 08/IN.1/I2103: Adaptive and adaptable media and services for dynamic personalisation and contextualisation (AMAS).

References

- [1] Altova Mapforce. <http://www.altova.com/mapforce.html>, 2012.
- [2] F. Abel, D. Heckmann, E. Herder, and J. Hidders. A framework for flexible user profile mashups. In *Workshop on Adaptation and Personalization for Web 2.0, International Conference on User Modeling, Adaptation and Personalization - UMAP 2009*, 2009.
- [3] L. Aroyo, P. Dolog, A. Naeve, M. Nilsson, and F. Wild. Interoperability in Personalized Adaptive Learning. *Educational Technology & Society*, 9(2):4–18, 2006.
- [4] M. Assad, D. Carmichael, J. Kay, and B. Kummerfeld. PersonisAD: Distributed, active, scrutable model framework for context-aware services. In *Pervasive Computing: Proceedings of the 5th International Conference - PERVASIVE 2007*, volume 4480 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2007.
- [5] S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci. Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07*. ACM, 2007.
- [6] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286, Nov. 2008.
- [7] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina. User Modeling in a Distributed E-Learning Architecture. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*. Springer, 2005.
- [8] F. Carmagnola, F. Cena, and C. Gena. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, 21(3):285–331, Feb. 2011.
- [9] F. Carmagnola and V. Dimitrova. An Evidence-Based Approach to Handle Semantic Heterogeneity in Interoperable Distributed User Models. In *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems - AH 2008*, volume 5149 of *Lecture Notes in Computer Science*, pages 73–82. Springer, 2008.
- [10] F. Cena. Integrating web service and semantic dialogue model for user models interoperability on the web. *Journal of Intelligent Information Systems*, 36(2):131–166, 2011.
- [11] P. Dolog and M. Schafer. A Framework for Browsing, Manipulating and Maintaining Interoperable Learner Profiles. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*. Springer, 2005.
- [12] S. Falconer and M.-A. Storey. A cognitive support framework for ontology mapping. In *The Semantic Web: Proceedings of the 6th International Semantic Web Conference - ISWC 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2007.
- [13] S. M. Falconer, N. Noy, and M.-a. Storey. Ontology mapping-a user survey. In *Ontology Matching Workshop, International Semantic Web Conference - ISWC 2007*, 2007.
- [14] D. Heckmann and A. Krueger. A user modeling markup language (UserML) for ubiquitous computing. In *Proceedings of the 9th International Conference on User Modeling - UM 2003*, volume 2702 of *Lecture Notes in Computer Science*, pages 148–152. Springer, 2003.
- [15] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. GUMO-the general user model ontology. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*, pages 428–432. Springer, 2005.
- [16] J. Kay, B. Kummerfeld, and P. Lauder. Personis: a server for user models. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems - AH 2002*, volume 2347 of *Lecture Notes in Computer Science*, pages 203–212. Springer, 2002.
- [17] B. Mehta, C. Niederee, A. Stewart, M. Degenmis, P. Lops, and G. Semeraro. Ontologically-Enriched Unified User Modeling for Cross-System Personalization. In *Proceedings of the 10th International Conference on User Modeling - UM 2005*, volume 3538 of *Lecture Notes in Computer Science*. Springer, 2005.
- [18] P. Shvaiko and J. Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2012.
- [19] 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*, 16(1):64–76, 2006.
- [20] E. Walsh, A. O'Connor, and V. Wade. Evaluation of a Domain-Aware Approach to User Model Interoperability. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media - HT 2012*, pages 197–206. ACM, 2012.
- [21] 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 *Proceedings of the 11th International Conference on User Modeling - UM 2007*, volume 4511 of *Lecture Notes in Computer Science*, pages 27–36. Springer, 2007.