Evaluation of a Domain-Aware Approach to User Model Interoperability

Eddie Walsh
Knowledge and Data Engineering Group
Trinity College Dublin
Dublin 2, Ireland
Eddie.Walsh@scss.tcd.ie

Alexander O'Connor
Knowledge and Data Engineering Group
Trinity College Dublin
Dublin 2, Ireland
Alex.OConnor @scss.tcd.ie

Vincent Wade
Knowledge and Data Engineering Group
Trinity College Dublin
Dublin 2, Ireland
Vincent.Wade@scss.tcd.ie

ABSTRACT

It is becoming increasingly important to facilitate the integrated management of user information. Exchanging user information across heterogeneous systems has many benefits, particularly in enhancing the quality and quantity of user information available for personalization. One common approach to user model interoperability is the use of mapping tools to manually build rich executable mappings between user models. A key problem with existing approaches is that the mapping tools are often too generic for these specialized tasks and do not provide any support to an administrator mapping in a specific domain such as user models. This paper presents a novel approach to user model interoperability which lowers the complexity and provides support to administrators in completing user model mappings. The domain-aware approach to user model interoperability incorporates interchangeable domain knowledge directly into the integration tools. This approach was implemented in a system called FUMES which is a mapping creation and execution environment that includes two domain-aware mechanisms; a canonical user model and user model mapping types. FUMES was deployed in an integration of existing user models and the domain-aware approach was then evaluated in a user study. The evaluation consisted of a direct comparison with a generic approach to user model interoperability which was applied using the commercial mapping tool, Altova Mapforce. The results of this evaluation demonstrate improvements in mapping accuracy and usability when using the domain-aware approach compared to the generic mapping approach.

Categories and Subject Descriptors

D.2.12 [**Software Engineering**]: Interoperability—*Data mapping*; D.2.11 [**Software Engineering**]: Software Architectures—*Domain-specific architectures*; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia

HT'12, June 25–28, 2012, Milwaukee, Wisconsin, USA.
Copyright 2012 ACM 978-1-4503-1335-3/12/06 ...$10.00.

General Terms

Design, Experimentation, Human Factors

Keywords

User Modeling, Integration, eLearning, Personalization

1. INTRODUCTION

A vast amount of information about users is now being accumulated within organizations and on the open web [13]. Within organizations such as those in the corporate, academic, medical or governmental domains, extensive user information is a central component of many systems. For example, in recent surveys 90% of responding companies use administration systems to manage human resources [24] and 79% have used or plan to use a learning management system to manage employee educational profiles [21]. Similarly, on the open web, users provide and manage their own information through the many services they use. For example, popular social networking websites such as Facebook and LinkedIn allow users to maintain extensive personal and professional profiles. In many cases, these various online systems and services are gathering information about the same user; however, this information is rarely connected and shared due to many organizational and technological boundaries [13]. Federating user information from multiple sources can potentially bring many benefits to enhance the user experience in software applications [3]. Richer and more dynamic models of the user can be constructed [20]. Sharing user information in this way can increase the knowledge of the user available to each system thereby allowing improved user-centric functionality, while at the same time reducing repetitive user interactions [29]. Potentially, the main benefit of user model federation is in the area of personalization. In this case, richer user models allow for more tailoring of content and services to the specific needs of each individual user.

As with all interoperability scenarios, there are major challenges in sharing user information effectively. Even if the user information is accessible, the heterogeneity of the various systems and their representations of user information can be significant at the structural, syntactical and semantic levels. Manual mapping techniques have often been adopted when attempting to overcome this heterogeneity [28]. Manual mapping consists of one or more mapping administrators [19] identifying and mapping equivalent attributes between data models often using custom-made translation scripts or one of the many generic mapping tools available such as Altova Mapforce [2].

However, as user models begin to exchange more expansive and detailed types of user information, the mappings that are required also become more specialized. For example, education-based user models could exchange information such as a user's identification details, social connections, learning preferences, course assessments and current competencies. This expansive user information increases the complexity of the manual mapping task requiring mapping administrators that possess increasingly specialized domain knowledge. Current generic mapping approaches and tools offer little support for domain-specific mapping tasks.

In this paper, the development and evaluation of a novel domain-aware approach to user model interoperability is presented. Domain-awareness is defined as the incorporation of domain-specific knowledge into generic tools and processes to support complex tasks in a particular domain. In this case, domain-awareness is the application of knowledge of the user domain to support the mapping of user models. The domain-aware approach is applied in the creation and execution of mappings in the form of two domain-aware mechanisms, a canonical user model and user model mapping types, which tailor the process to the user domain. The canonical user model is a consistent shared user model that all mappings are created to and the user model mapping types are mapping components specifically for creating mappings between user models. The application of the domain-aware approach in a mapping-based exchange system called FUMES is presented. Finally, a user-based evaluation of the domain-aware approach is presented which demonstrates improvements in mapping accuracy and usability when compared to a generic mapping approach.

2. RELATED WORK

A variety of approaches to provide user model interoperability have been attempted. These approaches have examined a number of key issues in this area such as the management of multiple user model exchange scenarios and the syntactic and semantic heterogeneity of user models.

The management of user model exchange between systems is generally performed using a centralized or distributed approach. In early systems, the completely centralized approach was the most common implementation in the form of the user model server [11] [23]. However, its inability to support self-contained systems led to variations on these designs that were more flexible [12] [20] [17]. Some designs for totally distributed approaches have been put forward but implementations are less common due to the complexity involved in the process [30]. Currently, the most common implementations are hybrid approaches that take aspects of both the distributed and centralized methods to create solutions that are less difficult to develop and can support independent, heterogeneous systems [28] [8] [14] [6].

When executing the exchange of user models, some of the approaches identified use a pre-runtime, administrator-initialized process [17]. Other approaches can perform the exchange in a runtime, on-demand process [28] [12] [20] [30]. In many of the approaches the exchange is performed using complete user models, however, some implement more complex exchange using user model fragments [20] [30] [17].

Most approaches provide support for multiple user models interoperability scenarios. An important aspect of this is the resolution of inconsistencies such as overlap and incompleteness of user information [3] [17]. Only a few of the approaches provide details on how they reconcile these inconsistencies [28] [8]. Some solutions include the removal of repeated and conflicting information through intelligent mappings [28]. These mappings can eliminate overlapping information based on dates or precedence. For the problem of incompleteness, no solutions were provided by these approaches. Some suggested that the exchange would sometimes result in partial user models, which few existing systems are capable of utilizing [28].

To provide syntactic interoperability of user models, the adoption of a common user model format and exchange protocol is the typical solution. Some of the research approaches use semantic web technologies such as RDF or OWL to represent the user model information [1] [28] [17] [8]. These languages provide more explicit semantics and can be used for additional benefits such as reasoning. However, the most common format to represent and extract user information from existing applications is XML [11] [16] [25]. For the transfer protocol, the most popular method is certainly the common adoption of web service technologies such as REST and SOAP [12] [8] [14].

To overcome semantic heterogeneity, standardization or mapping techniques are often used. Most of the approaches have adopted some form of a canonical model but these have been designed for different purposes. Several approaches use the canonical model as the standard to which every application should conform to provide interoperability [12] [20] [17]. This bypasses many of the semantic interoperability problems. However, many of these approaches have developed different canonical models limiting wider interoperability. Some approaches have used the canonical model as a basis to perform semantic mappings from the individual systems to a common location [28] [8]. In this way it is a means to reduce heterogeneity between different systems and control the number of mappings required [28]. This approach provides much greater potential for widespread interoperability between truly heterogeneous applications. Many approaches have developed canonical model ontologies that are based upon the main user model specifications such as IMS LIP [17], or else use GUMO [28] [20]. This, combined with the adoption of RDF or OWL as the user model format, has led to some of these approaches using ontology matching and mapping techniques.

However, few of the approaches that acknowledged the need for semantic mapping provide custom tools to aid the process. Semantic integration is generally provided by either manually writing custom integration scripts or by employing one of the current generic schema or ontology mapping tools available such as Altova Mapforce [2], COMA++ [4] and PROMPT [26]. Surveys of users of these tools have found them to be often too general and built without domain-specific mechanisms, lacking visual displays or easy to use tools, and not allowing for expressive enough mappings [18]. Some of the approaches mention that these tools provide semi-automation of the mapping process [28]. However, for the more heterogeneous user models this will likely provide limited benefits and a complex manual mapping stage will still be required [7].

Overall, these approaches are representative of the current level of progress in the field of user model interoperability.

One of the key issues that emerged was the lack of domain-specific tools to support the resolution of heterogeneity in the user models. Current mapping tools are not easy to use for this complex task and significant improvements in user model interoperability can be made by improving the integration tools and making them more suitable for domain-specific tasks.

3. DOMAIN-AWARE USER MODEL INTEROPERABILITY

The domain-aware approach incorporates domain-specific knowledge into generic tools and processes to support complex tasks in a particular domain. To apply this approach to user model interoperability, knowledge of the user domain is leveraged to support the mapping of the user models. The domain-aware approach is applied in the creation and execution of mappings in the form of two domain-aware mechanisms. These mechanisms, a canonical user model and user model mapping types, tailor the process to the user domain. The canonical user model is a shared user model representation that all mappings are created to and the user model mapping types are mapping components that are specifically designed for mapping user information and facilitating reuse. Both of these mechanisms were selected as they allow the addition of domain-awareness in key areas of the mapping process but do not significantly restrict the tool to a single domain; allowing rapid interchangeability to reuse the tools in other domains.

Some examples of other potential domain-aware mechanisms include a mapping visualization designed specifically for user model information or a specialized user model matching algorithm that derives suggestions for mappings between user models. However, in contrast with the two chosen mechanisms, these mechanisms would restrict the tool to the user model domain and require significant development effort to apply the tool in other domains. In the following sections, both of the chosen domain-aware mechanisms are described in more detail.

3.1 Canonical User Model

As a single user model standard has yet to gain widespread adoption, much research has been conducted into other methods to overcome heterogeneity. One approach that is common in other areas, such as the federation of database systems, is the provision of mappings. Mapping consists of creating direct associations between equivalent elements of individual data models. Figure 1 shows the direct mapping approach using four user models as an example. In this case, mappings are created from user model 1 to user model 2, 3 and 4 directly. Similarly, the remaining user models are all mapped to each other in this approach. For data translation, these mappings can be executed to transform the relevant data between different user model representations.

A variation of this approach takes aspects of both standardization and mapping to provide canonical model mapping. In this form, mappings are created from each user model solely to a canonical model, which provides a common representation of user model concepts. This approach is also shown in Figure 1. For data translation, the mappings can be executed to transform data into a canonical representation before transforming into the chosen target user model representation.

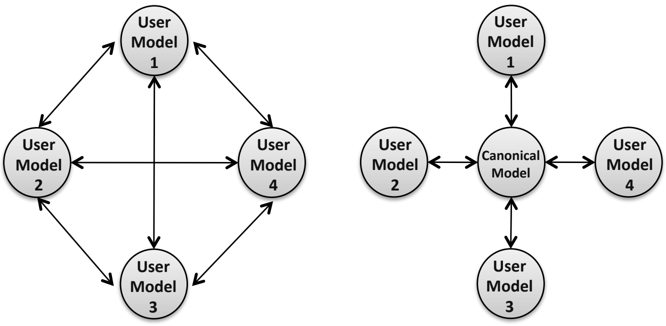


Figure 1: Direct and canonical mapping approaches

There are advantages and disadvantages to both approaches. One of the key issues is the quantity of mappings required by each approach. In the direct mapping approach, each user model is directly mapped to every other. Each time a new system is added to an interoperability scenario the number of bi-directional mappings sets required will correspond to the number of previously mapped systems. Thus, the total number of mapping sets required will increase every time a new system is added. The canonical model mapping approach results in the number of mapping sets being controlled. Each time a new application is added it only requires one bi-directional mapping to the canonical model. It is then, as a result of previous mappings to the canonical model, mapped to every other application automatically.

The quantity of mappings required raises a number of related issues. If the mappings are manually created the workload for an administrator is less when using a canonical model mapping approach rather than a direct mapping approach. The maintenance of the mappings is also affected. If the source user models are edited, the number of mappings to be changed is less in the canonical model mapping approach than in the direct mapping approach. The performance of the execution of the mappings can also be impacted upon by the quantity of mappings. As a result, the direct mapping approach is potentially slower than the canonical model approach to execute mappings for data exchange purposes.

However, the canonical model mapping approach produces new issues not present in the direct mapping approach. As a complete user model representation is infeasible, the canonical model must be editable and extensible to support a wide variety of user information. This raises the issues of evolution and management of the canonical model over time. Changes to the structure of the canonical model could potentially require changes to multiple existing mappings. This requirement for extensibility could also be a benefit of the canonical model approach as it provides more flexibility and is not as constrained as a standardized user model such as the IMS Learner Information Package (LIP) [22].

This research is focused on the application of domain-aware mechanisms to the creation and execution of mappings for the purpose of user model interoperability. As a result, the canonical model approach has been adopted for a number of key reasons.

In the creation of mappings, the canonical model approach can be used within a manual mapping tool. Using a canonical model provides consistency across multiple mapping creation scenarios and as a result can allow the visualization to be clearer and easier to navigate. This can potentially improve the manual mapping process for the administrator.

In the execution of mappings, the use of a canonical model also provides benefits in the automatic exchange of user models. The canonical model provides control of the number of mappings required in multi-system interoperability scenarios. It potentially makes the integration of a new user model much easier as a single mapping set to the canonical model is required. The lower number of mappings could also improve the overall performance of the user model interoperability.

3.2 User Model Mapping Types

Many generic mapping types have been developed to overcome heterogeneity issues and are commonly used in many mapping tools [2]. However to date, no mapping types have been developed that focus on common mapping problems in a specific domain such as user models. The development of user model mapping types is another method where domain-awareness can be applied to a commonly generic process.

From examining sample user information from a variety of educational web systems such as Sakai [27], Moodle [25], AHA! [16], CUMULATE [11] and APeLS [15] a set of common user model mapping types in the educational domain have been derived. These user model mapping types consist of core generic mapping types, which are equivalent to those in many other generic integration tools such as Mapforce, but they are combined with domain-specific information to provide an administrator with a selection of mappings that are specifically for heterogeneous user models. In the following sections, some of the most common generic mapping types are explained and in Table 1 examples of how they can be used to form user model mapping types are given.

Schema mappings are the most basic form of generic mapping and are created between the equivalent schema elements of two user models. Instance mappings allow more complex mapping of specific instance data from user model elements. Functional mappings allow generic manipulation of instance data in an exchange between user model schema elements. Types include numeric mappings which allow mathematical manipulation of numerical data, format conversions which allow manipulation of data types such as dates, and interval mappings which allow manipulation of data that contains numeric intervals.

These mapping types can also represent many-to-one, one-to-many or many-to-many relationships. For example, the schema mapping can be joined, where multiple schema elements from one user model are equivalent to one schema element in another user model, and separated, where one schema element is equivalent to multiple schema elements. Multiple instance values can also be defined as equivalent within a single mapping.

Similarly to existing integration tools, these generic mapping types can be used to construct mappings between user models. The administrator must construct the entire mapping from the start each time, identifying every required value and have a complete understanding of how it should function. However, the generic mapping types can be combined and stored with domain-specific information to create user model mapping types that are more specialized for mapping between different categories of user information. In a web-based system, these user model mapping types can be reused by the administrator or shared with other administrators to build a large collection of relevant mappings in the domain of interest.

A basic example of a user model mapping, as can be seen in Table 1, would be the conversion of a numeric grade to a text-based grade between user models where 11-20 = Pass and 0-10 = Fail. If this mapping does not already exist, it could be created using the generic interval mapping and saved as a user model mapping type, called for example ”Convert grades from 0-20 to Pass/Fail”. This domain-specific mapping would then be available to be reused any time this conversion was required in a user model. The administrator does not have to construct the entire mapping and can just search and select it from a list of relevant user model mappings.

Overall, the domain-aware approach to mapping has many benefits. Potentially, there would be a large amount of domain-specific mappings created in an area such as user models. In a web-based system, there is significant potential for mapping reuse, collaborative mapping and improving automatic matching based on the analysis of existing approved mappings. These domain-aware mapping types should be more relevant and easier to find and implement by an administrator who is not an expert in mapping techniques.

The generic mapping types and subsequent user model mapping types were identified using an evidence-based approach where existing education-based user models were analyzed for potentially shareable information. The selected systems are a representative sample of typical educational web systems and use many common user modeling techniques, such as the overlay approach in adaptive systems [10], that would potentially be present in other systems' user models. More mapping types are likely to be included as the analysis is expanded to other types of user data.

4. FEDERATED USER MODEL EXCHANGE SERVICE (FUMES)

Providing a means to analyze the domain-aware approach has led to the development of an interoperability system called the Federated User Model Exchange Service (FUMES) [31], shown in Figure 2. FUMES combines an administrator-led mapping creation stage with an automatic mapping execution stage to provide a comprehensive user model interoperability process. To support both stages FUMES includes two key components, the Mapping Tool and the Translation Service. Both of these components incorporate the two domain-aware mechanisms, the canonical user model and the user model mapping types, to provide greater support in the complex task of mapping heterogeneous user models.

The Mapping Tool is a graphical tool for the manual creation of mappings between user models and is shown in Figure 3. The administrator can use this tool to perform a number of tasks such as the graphical creation and testing of user model mappings. Both domain-aware mechanisms are supported within the Mapping Tool. The canonical model is used in the visual creation of the mappings. This is fundamentally different to the majority of existing mapping tools as it provides a clearer and more consistent visualization, allowing the administrator to continually map to the same user model structure. The canonical user model is based on the IMS Learner Information Package (LIP) specification

Table 1: Mapping Type Examples

Generic Mapping Type	Example
Schema equivalence	value1 = valueA
Schema join	value1 + value2 = valueA
Schema split	value1 = valueA + valueB + valueC
Instance multiple	value1[value2, value3] = valueA[valueB,valueC]
Instance join	value1[value2] + value3[value4] = valueA[valueB]
Instance split	value1[value2] = valueA[valueB] + valueC[valueD]
Numeric	value1[value2] * 100 = valueA[valueB]
Interval	value1(value2-value3) = valueA[valueB]

User Model Mapping Type	Example
Basic name equivalence	firstname = forename
Join first name & last name	firstname + lastname = fullname
Split address into sections	address = street + city + country
Convert user ids	userid[jsmith, rjones] = userid[06125,00242]
Join SQL concepts	concept[SQL1] + concept[SQL2] = concept[SQLA]
Split SQL concepts	concept[SQL1] = concept[SQLA] + concept[SQLB]
Multiply grade by 100	score[0.8] * 100 = result[80]
Convert grades from 0-20 to Pass/Fail	score[11-20,0-10] = grade[Pass,Fail]

[22]. The user model mapping types are also used in the visual creation of mappings, allowing the administrator to generate and reuse specialized user model mappings more quickly. The mappings created by the Mapping Tool are stored in the FUMES database for later use in exchanging user models between systems. The mappings are saved as XQuery, a powerful query and translation language, which allows for the easy execution of mappings to exchange between XML-based user models.

When the administrator-led mapping creation stage has been completed for a number of systems, FUMES can perform a mapping execution stage to automatically exchange user information between those systems. The central point for exchange is the Translation Service. This service handles the management of the user model interchange and translates between the various user model representations. The Translation Service uses the web service approaches REST and SOAP to allow access to heterogeneous user models and provides a means to transfer them between different systems using common technological standards. Currently, the Translation Service supports user models represented in XML; the most commonly used format in existing web systems. In the future, case study integrations will be conducted with user models represented using semantic web technologies such as RDF and OWL. The two domain-aware mechanisms are also supported in the Translation Service. The heterogeneous user models are translated into the common canonical user model representation during exchange scenarios allowing greater control of the mapping execution. The user model mappings also allow for easier maintenance and potential performance gains, as fewer mappings are required compared to traditional mapping systems.

To date, all of the main aspects of FUMES have been developed and deployed successfully in integration scenarios with existing systems. Future work will address some key challenges in the provision of long-term user model interoperability using the domain-aware approach, specifically focusing on sharing, reuse and collaboration within the system. One of the key challenges of applying the domain-aware approach is that the required mechanisms, such as the canonical model, may not be available for new domains and will need to be developed prior to using domain-aware

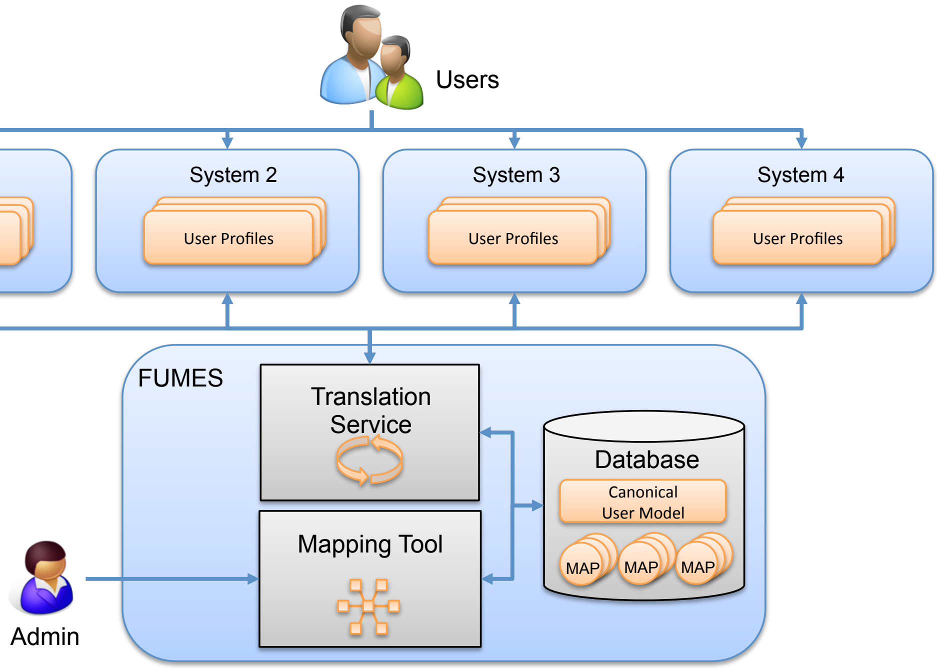


Figure 2: FUMES Architecture

integration tools. Making this process quicker and easier for administrators will be a key focus of future development of the tool, however, the adoption of web-based technologies allows administrators to easily share and reuse different canonical models. Similarly, the user model mappings can be reused by the administrator or shared with other administrators to build a large collection of relevant mappings in the domain of interest. Also, as mentioned in Section 3.1, editing and extending the canonical model is important in the integration of multiple heterogeneous user models and greater support for this will be provided in future versions of the tools, including the automatic editing of related existing mappings.

5. EVALUATION OF DOMAIN-AWARE USER MODEL INTEROPERABILITY

In the following section, a user-based evaluation of the domain-aware approach to user model interoperability is presented, consisting of the experimental setup used and results.

5.1 Experimental Setup

The evaluation of the domain-aware approach to user model interoperability consisted of a comparison of FUMES, which contains the domain-aware mechanisms, and the commercial mapping tool Altova Mapforce, in which a typical generic mapping approach was applied. Experiment participants were required to perform the mapping administrator role and complete four mapping tasks using both FUMES and Mapforce. The participants conducted the experiment individually and the order in which they used the tools was alternated to account for any learning of the tasks during the experiment. The evaluation was conducted initially with 12 participants who were all technically proficient and had limited experience in user models and mapping techniques. The overall experiment duration was two hours, one hour for the participant to use each tool. This included 20 minutes basic training in the tool using example mappings, 30 minutes completing the mapping tasks, and 10 minutes completing the feedback questionnaires.

The mapping tasks were chosen to represent typical mappings between multiple heterogeneous systems. The mapping tasks and user models were derived from existing systems and focus on general identification, assessment and competency information in the area of SQL. The four user models were retrieved from Sakai [27] and Moodle [25] which are the two most popular open source Learning Management Systems (LMS), an adaptive SQL web course based on the APeLS system [15], and a web-based user modeling system, CUMULATE [11]. The mapping tasks increase in difficultly with the third and fourth tasks being more complex and requiring more domain-specific knowledge than the first and second tasks. In order to manage the total duration of the experiment it was necessary to have a time limit associated with each task. The first and second tasks were limited to five minutes (300 seconds) and the third and fourth tasks were limited to ten minutes (600 seconds). When the time limit was reached the participants were asked to continue to the next task. The mapping tasks are described in the following sections.

Task 1 *Map forename in UM1 to its equivalent in UM4.*

This mapping task was the most basic and consisted of a direct equivalence mapping between two equal elements.

Task 2 *Map the country from address in UM3 to its equivalent in UM4.*

This mapping task was a more difficult mapping that required the tokenization of an address string and extraction of the country segment.

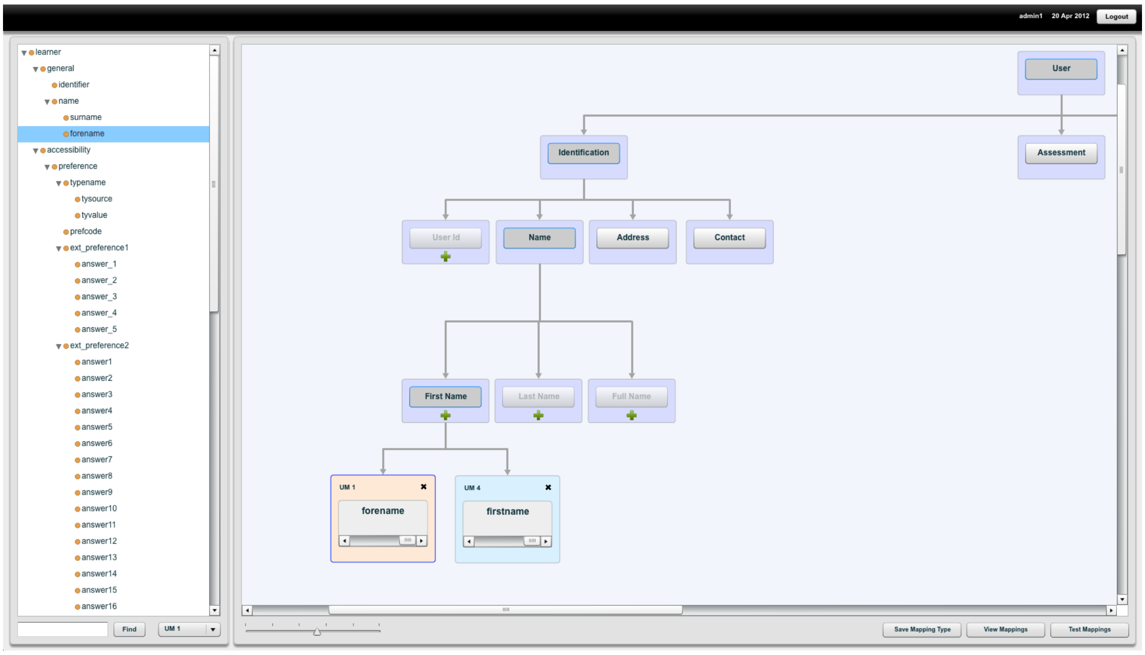


Figure 3: FUMES Mapping Tool

Task 3 *Map the finalscore for "Final Exam" in "Advanced Database Systems" in UM3 to its equivalent in UM4. Use the following to convert the finalscore: 11-20 = Pass, 0-10 = Fail.*

This mapping task was significantly more complex and involved the identification of a numeric grade within a course and its conversion to an equivalent text-based course grade.

Task 4 *Map the concept "db.tables.populate.insert" in UM1 to its equivalent numeric knowledge value in UM2. Use the following to convert the concept: concept exists = 0, concept does not exist = 1.*

This mapping task was the most complex and involved checking for the existence of a text-based competency and its conversion to a numeric knowledge value.

The mapping tasks represented the identification, creation and verification of executable mappings between existing systems that could be used in a real user model exchange scenario. Each of these tasks required the participants to examine the user models, identify the appropriate section of the source and target user models, select the appropriate mapping type, and construct and test the mapping.

When using FUMES the participants used the domain-aware approach represented by the domain-aware mechanisms, the canonical user model and user model mapping types. The canonical model in the experiment is based on the IMS LIP specification [22] and represents general identification, assessment and competency information in the area of SQL. To complete the tasks, 25 user model mapping types were provided in FUMES which represented typical mappings in user models in the area of identification, assessment and competencies. In Mapforce, the participants used a generic approach and mapped directly between the user models using a selection of 10 relevant generic mapping types to construct the mapping tasks.

5.2 Results

In the following section, the results from the experiment are presented and analyzed. This includes an analysis of the accuracy and duration of the mapping tasks and an analysis of the overall usability of the mapping tool.

5.2.1 Mapping Accuracy and Duration

In this section the results of the mapping tasks in both FUMES and Mapforce are compared for accuracy and duration.

To analyze the accuracy a gold standard for each mapping task was created. The gold standard consisted of the various stages involved in successfully completing the mappings in both tools. For each stage of each task an accuracy score was given; 0 for incomplete or major errors, 0.5 for minor errors or 1 for completed stages. The overall mean of these stages was then used to calculate a percentage accuracy score for each individual task and for the tasks overall.

As can been seen in Figure 4, task 1 had 100% accuracy in both tools. This demonstrates that users can perform basic equivalence mappings in both tools equally well. For task 2, there is a difference in accuracy of 33%, in task 3 it is 30% and in task 4 it is 20%. The decline in the difference could indicate that it takes longer to learn how to map accurately using the generic approach in Mapforce.

There was an overall difference in accuracy across all tasks of 29%. Paired t-tests were carried out on the accuracy results and showed there is a statistically significant difference in accuracy ($p<0.001$) between the domain-aware approach in FUMES and the generic approach using Mapforce. These results represent the mapping task accuracy within the specified time limits. It is possible that higher levels of accuracy would be achievable in these tools if there were no time limits applied.

To analyze the duration of the mapping tasks, the average time taken in both FUMES and Mapforce to perform each of

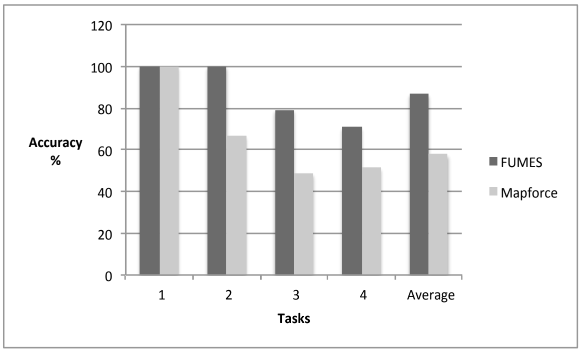


Figure 4: Task Accuracy Results

the mapping tasks was recorded. Each task had a time limit to control the overall duration of the experiment. The first and second tasks were limited to five minutes (300 seconds) and the third and fourth tasks were limited to ten minutes (600 seconds).

As can be seen in Figure 5, the first task is completed much faster in Mapforce (47% of the allotted time in FUMES vs 26% in Mapforce). The likely reason for this is because, in Mapforce, no complex mapping components were required for this task and the mapping could be created using a simple line connector between the equivalent parts of the user model. FUMES uses the canonical model mapping approach which, in these tasks, requires two mappings to the canonical model for every one direct mapping in generic approach. Task 2 is completed faster in FUMES (68% of the allotted time in FUMES vs 96% in Mapforce). This indicates that participants took longer to learn how to use the generic mapping types in Mapforce. Durations for task 3 (90% of the allotted time in FUMES vs 89% in Mapforce) and task 4 (81% of the allotted time in FUMES vs 90% in Mapforce) are much closer indicating that the most complex mappings were difficult to fully complete in both tools.

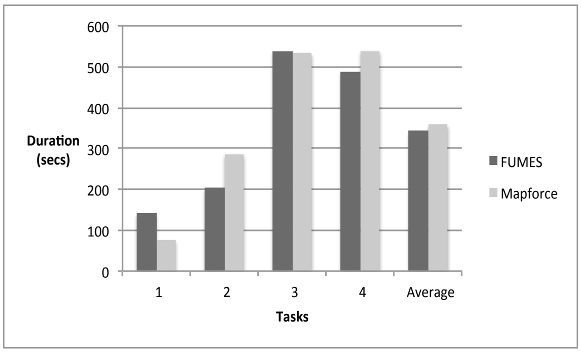


Figure 5: Task Duration Results

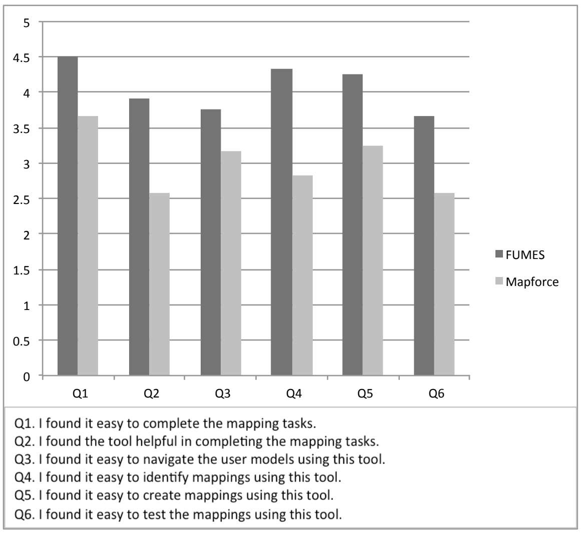
The average time spent on all the mapping tasks in FUMES is 76% of the maximum allotted time compared to 80% in Mapforce. Paired t-tests show that this is not a statistically significant difference ($p>0.05$) so we can not conclude that mapping is faster using the domain-aware approach. However, a similar amount of time is required to perform the mapping task in both tools. This result is important because, as described in Section 3.1, the canonical model approach requires more mappings than the direct approach initially. However, as the number of systems increases, the canonical approach will require less mappings than the direct approach. Therefore the canonical model approach would almost certainly be faster than the direct approach when creating larger numbers of mappings between user models.

5.2.2 *Mapping Tool Usability*

Following the analysis of the accuracy and duration of the mapping tasks, the overall usability of both tools was analyzed. This was a questionnaire-based analysis consisting of a System Usability Scale (SUS) [9] study followed by a series of custom task and tool related questions.

SUS is a widely used and reliable questionnaire tool for measuring the usability of a variety of products and services. It generates a single value that can be used to determine the usability of an individual system and to compare the usability of multiple systems. To determine the relative value of the SUS score an adjective rating scale can be applied [5]. FUMES obtained a SUS score of 72 that indicates a ”good” level of usability and Mapforce obtained a score of 50 that indicates ”ok” usability.

Following the SUS study, the participants were asked a number of custom task and tool related questions. The responses are summarized in Figure 6. For all of these questions a more positive response was recorded when using FUMES compared to Mapforce. The participants stated that it was easier to complete the tasks in FUMES and that they found FUMES more helpful. They also found it easier to navigate the user models, and to identify, create and test mappings using FUMES.



Q1. I found it easy to complete the mapping tasks.
Q2. I found the tool helpful in completing the mapping tasks.
Q3. I found it easy to navigate the user models using this tool.
Q4. I found it easy to identify mappings using this tool.
Q5. I found it easy to create mappings using this tool.
Q6. I found it easy to test the mappings using this tool.

Figure 6: Task Related Results

To gain further understanding of how the participants viewed both tools a number of open questions were also presented to the participants. The participants were asked which aspects of the tools they liked. For FUMES the most common responses were the general ease-of-use of the user interface and the ability to use user model mapping types. For Mapforce, the most common responses were the navigation and drag and drop functionality and the ability to create and link mapping types.

The participants were also asked what aspects of the tools they disliked. In this case, the most common responses for FUMES were related to the layout, buttons and xml viewing in the testing area. For Mapforce, the most common responses were selecting mapping types and creating mappings and the cluttered user interface when there are multiple mappings between user models.

Finally, participants were asked to directly compare the mapping approaches (i.e. direct mapping or canonical model mapping) and mapping creation (i.e. generic mapping types or user model mapping types) of the two tools. 67% of participants stated that they found the FUMES mapping approach easier to use. The most common comments were that it was easier to create mappings in the user model domain using the canonical model. However, some noted the domain-specific nature of the user interface and questioned its applicability to other domains. 33% of participants also stated that they preferred the direct mapping approach with most comments stating that they found that approach more intuitive. 100% of participants stated that they found it easier to create mappings in FUMES. The most common comments were that the user model mappings were more relevant and easier to use.

6. DISCUSSION

The results provided by the user study have generated a number of interesting insights into the impact of the domain-aware approach to user model interoperability.

Mapforce was chosen to deploy a generic mapping approach in this experiment as it was one of the best mapping tools available and one of the few tools which could create and execute the required complex mappings. Its graphical interface was also more independent of any translation language and did not require the administrator to have specific skills to start mapping. Mapforce is a large commercial tool and contains many more features than were used in this experiment. However, the results of the experiment demonstrate that for mapping tasks in a restricted domain, the domain-aware approach in FUMES performs better than the generic approach when using Mapforce.

The domain-aware approach in FUMES provides a significant increase in accuracy and has similar mapping duration except in the case of the most basic equivalence mappings. Many of these basic equivalence mappings could also potentially be completed through the introduction of an automatic matching tool which will be incorporated into FUMES in the future. Across the tasks, there is a decrease in the accuracy difference between tools indicating that Mapforce may take longer to learn how to use. There is a significant increase in duration of mappings from more simple mappings to more complex mappings. The more complex mappings took approximately three times longer than the more basic mappings. This suggests that mapping tools should provide more support in the completion of complex mappings.

The domain-aware approach in FUMES also provides better usability than the generic approach in Mapforce. This was demonstrated using both the SUS questionnaire and the tool and task related questions. It is possible that some of the recorded improvement in usability was as a result of the general layout and appearance of the user interface. However, this is likely to be negligible as the most significant differences in the design of FUMES when compared to Mapforce result from the use of the domain-aware mechanisms.

While the mapping accuracy and duration analysis was focused on assessing the overall impact of the domain-aware approach on user model mapping, the usability study also gave some insights into the impact of the individual domain-aware mechanisms. The answers and comments from the participants clearly state that the user model mapping types were preferred to the generic mapping types although some users preferred the flexibility of linking different mapping types in Mapforce. The use of the canonical model mapping approach was also preferred but not as conclusively. One third of users preferred the direct mapping approach used in Mapforce which suggests that further analysis may be required in order to determine if the canonical model approach is suitable for all users in all interoperability scenarios.

Overall, the evaluation has provided interesting insights into many aspects of the domain-aware approach. However, there are also other areas where experimentation could be performed in the future to examine the approach in further detail. These include examining the performance of users with varying skill sets, investigating the evolution of the domain-aware mechanisms over time with a small group of long-term users, and examining use in other domains.

7. CONCLUSIONS

This paper has described the development and evaluation of a novel approach to user model interoperability called the domain-aware approach. This approach differs from existing generic approaches by incorporating domain knowledge in tools and processes to support complex mapping tasks in specific domains. For user model interoperability, this approach is encapsulated in a system called FUMES. FUMES supports the creation of mappings between heterogeneous user models and the execution of those mappings to exchange user models between multiple systems. The domain-aware approach is provided in FUMES through two mechanisms: a canonical user model and user model mapping types. Both of these components provide domain knowledge that can support an administrator in the complex task of mapping user models.

The evaluation of the domain-aware approach in FUMES consisted of a direct comparison with a generic approach to user model mapping using an existing commercial mapping tool, Altova Mapforce. This evaluation took the form of a user study which examined both tools for mapping accuracy, duration and overall usability. The results of the evaluation were presented and they demonstrated that the domain-aware approach provides significant improvements in accuracy and usability while maintaining the speed of the majority of the mapping process. Overall, the results demonstrate that incorporating domain knowledge within the tools and processes allows administrators to perform better in what is typically the very complex and difficult task of mapping user models.

8. ACKNOWLEDGMENTS

This work is conducted within the AMAS (Adaptive Media and Services for Dynamic Personalization) research project which is funded by Science Foundation Ireland via grant 08/IN.1/I2103. The goal of AMAS is to research and develop innovative techniques and technologies to support dynamic, integrated adaptivity and personalization of web media and services to enable rich repurposing of existing content and web services.

9. REFERENCES

[1] 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 Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 16–27. Leibnitz University, Springer, 2010.

[2] Altova. Mapforce. http://www.altova.com/mapforce.html, 2011.

[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] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm. Schema and ontology matching with COMA++. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD '05*, page 906, New York, USA, 2005. ACM Press.

[5] A. Bangor, T. Staff, P. Kortum, and J. Miller. Determining What Individual SUS Scores Mean : Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3):114–123, 2009.

[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. 2007.

[7] P. A. Bernstein and S. Melnik. Model management 2.0: Manipulating Richer Mappings. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD '07*, page 1, New York, New York, USA, 2007. ACM Press.

[8] M. Bielikova and J. Kuruc. Sharing user models for adaptive hypermedia applications. *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pages 506–511, 2005.

[9] J. Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, Sept. 1996.

[10] P. Brusilovsky and E. Millán. User models for adaptive hypermedia and adaptive educational systems. *The adaptive web*, pages 3–53, 2007.

[11] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina. User modeling in a distributed e-learning architecture. *User Modeling 2005*, (0310576):387–391, 2005.

[12] P. Brusilovsky, S. Sosnovsky, and M. Yudelson. Ontology-based framework for user model interoperability in distributed learning environments. In *World Conference on E-Learning, E-Learn*, pages 2851–2855, 2005.

[13] F. Carmagnola, F. Cena, and C. Gena. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, 21(3):285–331, Feb. 2011.

[14] F. Cena and R. Furnari. A soa-based framework to support user model interoperability. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 284–287. Springer, 2008.

[15] O. Conlan and V. Wade. Evaluation of APeLS - An Adaptive eLearning Service Based on the Multi-Model, Metadata-Driven Approach. In P. De Bra and W. Nejdl, editors, *Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004)*, pages 291–295. Springer Berlin / Heidelberg, 2004.

[16] P. De Bra, A. Aerts, B. Berden, B. De Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash. AHA! The adaptive hypermedia architecture. *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia HYPERTEXT 03*, 4(1):81–84, 2003.

[17] P. Dolog and M. Schäfer. Learner modeling on the semantic web. In *Proceedings of Personalisation on the Semantic Web Workshop (PerSWeb) at the 10th International Conference on User Modelling 2005*, volume 5, Edinburgh, Scotland, 2005.

[18] S. Falconer, N. Noy, and M.-a. Storey. Ontology mapping-a user survey. In *Proceedings of the Workshop on Ontology Matching (OM2007) at ISWC/ASWC2007, Busan, South Korea*, 2007.

[19] B. Haslhofer. A Comparative Study of Mapping Solutions for Enabling Metadata Interoperability, 2008.

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

[21] IMC (UK) Learning Ltd. Learning Management Systems - are organisations making the most of them? Technical report, IMC (UK) Learning Ltd, 2010.

[22] IMS Global Learning Consortium. IMS Learner Information Package Specification. http://www.imsglobal.org/profiles/, 2011.

[23] J. Kay and B. Kummerfeld. Personis: a server for user models. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 203–212, 2002.

[24] L. Martin. CedarCrestone 2010-2011 HR Systems Survey Highlights. Technical report, 2011.

[25] Moodle. Moodle. http://moodle.org/, 2011.

[26] N. F. Noy and M. A. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, Dec. 2003.

[27] Sakai. Sakai. http://sakaiproject.org/, 2011.

[28] K. Van Der Sluijs and G.-j. Houben. A generic component for exchanging user models between web-based systems. *International Journal of Continuing Education and Lifelong Learning*, 16:64–76, 2006.

[29] J. Vassileva. Distributed user modelling for universal information access. *International Journal of Human-Computer Interaction*, pages 122–126, 2001.

[30] J. Vassileva, G. McCalla, and J. Greer. Multi-agent multi-user modeling in I-Help. *User Modeling and User-Adapted Interaction*, 13(1):179–210, 2002.

[31] E. Walsh, A. O'Connor, and V. Wade. Supporting Learner Model Exchange in Educational Web Systems. *7th International Conference on Web Information Systems and Technologies*, 2011.