Opening Learner Profiles across Heterogeneous Applications

Triomphé Ramandalahy, Philippe Vidal, and Julien Broisin
Institut de Recherche en Informatique de Toulouse
{ramanda,vidal,broisin}@irit.fr

Abstract

The more learner information is shared across a wide range of heterogeneous applications and tools, the more learner profiles will be relevant, reusable and useful for many systems and environments. This proposal stands on a widely used management standard, and introduces an open and generic learner profile supported by a service-oriented architecture ensuring exchange and reuse of learner information. The profile is characterized by (1) a core model integrating the LIP standard but also some metacognitive properties, and (2) a high abstraction level offering the possibility to extend the core model. The architecture facilitates sharing and reusing of learner profiles while providing scalability. Two use cases are presented and demonstrate (1) how heterogeneous applications can transparently collaborate to strengthen shared learner profiles, and (2) how resulting profiles can easily be consulted and extracted for further exploitation.

1. Introduction

Technology Enhanced Learning (TEL) leverages technology to maximize learning experiments within an educational environment that can offer learners the options of time, place and pace, but also facilitates elaboration of personalized learning styles [15]. To achieve this goal, a lot of researchers try to exploit learner profiles in order to allow teachers to draw conclusions about their students’ learning curve [7][10], or to bring adaptations to the learning scenario according to a given learner [1][8].

A learner profile can be defined as information describing a learner or a group of learners, collected or deduced from one or several pedagogical activities, computerized or not. Information specified in a learner profile may concern his/her knowledge, abilities, conceptions or behavior [4][6]. Since it is essential to enhance interoperability and scalability of learner profiles between different systems [14], the IEEE Personal and Private Information (PAPI) and the IMS Learner Information Package (LIP) [5] standards emerged. On one hand, these standards hinder the process of sharing and reusing learner profiles because it is very difficult for a client application to exploit learners profiles stored within various servers. On the other hand, their capability of extension is too low: even if LIP takes into account learner and producer profiles, it doesn’t allow to extending the schema with information describing, as an example, a tutor profile.

We present in this paper an open and generic learner profile able to integrate any existing or upcoming profiles, associated with a Service-Oriented Architecture (SOA) that facilitates sharing and reusing of learner information. This framework is based on the widely adopted WBEM (Web-Based Enterprise Management) architecture [2]; this standard brings a solution to unify management of distributed computing environments, and facilitates exchange of data across otherwise disparate technologies and platforms. The high abstraction level of the profile schema offers the opportunity to extend the core learner profile in order to meet requirements of a very specific learning application, or even to model a profile describing other user roles. The support architecture comprises an open source application for storing and managing profiles, together with a set of components and applications offering different services; some of them provide client applications and users with simple interfaces to discover and extend the profile schema, while others are responsible for importing data into and extracting data from the profile repository.

The next section briefly exposes the main WBEM features that justify our approach, and enters into details of the open learner profile to demonstrate how it can be extended to meet the requirements of a specific pedagogical purpose. Section 3 depicts the distributed architecture that facilitates sharing and reusing of learner profiles across heterogeneous applications. Then we focus on the benefits of our proposal by comparing it with popular existing standards and initiatives. Conclusions and further works are finally discussed at the end of the paper.
2. Design of an open learner profile

2.1 A standardized model driven approach

WBEM is based on a model-driven approach and exploits object concepts like classes, attributes, associations or inheritance for modeling of systems, networks and applications to supervise. To support these models, WBEM adopts a distributed architecture of management components.

The Common Information Model (CIM) is a metamodel composed of three main schema representing information to supervise. Since one of them offers the opportunity to define additional classes dedicated to a given environment, this metamodel represents an appropriate choice for representing an open learner profile. Indeed, its high abstraction level allows modeling of a wide range of specificities, including learners’ information. Moreover, the WBEM architecture is based on Internet technologies to ensure scalability and interoperability, thus facilitating sharing and reusing of data specified into the schema.

The open learner profile based on existing CIM models is presented in the next section, whereas the distributed architecture based on WBEM specifications is detailed in section 3.

2.2 Modeling of the core learner profile

The CIM User model [3] suggests two main classes to describe users: the class CIM_Person specifies administration attributes such as the user name, given name, or the mail address, whereas the class CIM_OtherPersonInformation provides details such as the preferred language or the user password. However, the CIM User model neither focuses on the pedagogical point of view, nor on the learner point of view. Therefore, we introduced the abstraction TEL_ProfileCore that represents the top-level class to design profiles. It ensures extensibility and openness, and covers any profile that may be required to optimize any TEL application or system.

Figure 1 depicts the core learner profile represented by the class TEL_LearnerCore. For interoperability reasons, this model ensures a full compatibility with LIP and separates a learner profile into four sub-profiles.

The Identification profile relates to administrative information of the learner and integrates attributes specified within the category Identification of LIP.

The Cognitive profile measures learner performances, goals and competencies. This sub-profile specifies most of the categories defined by LIP as an enumeration of associative arrays.

The Metacognitive profile aims to measure how a learner thinks about his/her cognitive skills. Learners who cannot monitor accurately cannot correct errors and as a consequence, they process information less efficiently than self-monitored learners.

The Preference profile includes three LIP categories and details information about what a learner would prefer to be applied during a distant learning session or about his/her general interests.

The next section specifies two services to easily exploit the extensible feature of the learner schema on one hand, and to manage the profile specific to a given learner on the other hand.

3. Architecture and use cases

3.1. Support components

Figure 2 illustrates the global architecture and components for sharing and reusing learner profiles. To support CIM models, our framework integrates Open Pegasus [12], a WBEM-compliant open source software that performs better than others existing tools [16]. It provides both a CIM repository called learner profile repository on Figure 2, and a CIM Manager responsible for safety and integrity of data. The open learner profile depicted on Figure 1 has been mapped to the CIM format, and the resulting files have been inserted into the repository.

The profile management service offers an overview of the current schema describing learner profiles, but also exposes methods to modify this data model by inserting, updating or removing classes and attributes of the core learner profile. The learner management service doesn’t aim to extend the core learner profile,
instead it allows any applications to retrieve and update the profile of a specific learner.

The management graphical interface illustrated on Figure 3 is a web application interacting with both management services. One can extend or modify the learner profile schema, and/or update the profile of a given learner. Everyone can get both the schema and the learner profiles stored into the repository, but for integrity reasons, operations of modification are restricted to authorized users.

Figure 2. An SOA for sharing and reusing learner profiles

Finally, two heterogeneous applications push learner information to the repository. An agent specific to the learning management system Moodle [11] is able to capture learner information stored into this system and to forward data to the learner management service. The other delivery system is a web testing system that helps to evaluate metacognitive properties of a learner. These concrete use cases are presented in the following section.

3.2. Use cases

The learning management system Moodle is the core component of the Technical Institute of the Toulouse University. Since various properties specified into the core learner profile can easily be extracted from information describing a user within the Moodle environment, we developed an agent able to capture the following learner information from this system:

- First name, last name and email address are integrated into the class CIM_Person.
- Description, encoded password, institution and preferred language match with some attributes of the class CIM_OtherPersonInformation.
- Interests are stored within the preference sub-profile.

Starting from others data describing a training or curriculum, additional information can be collected or computed and then mapped to attributes of the core learner profile: relationships are automatically gathered according to the teachers involved in the learner curriculums and stored into the preference sub-profile, whereas competencies match with the curriculum a learner is participating in.

Once the agent has collected all information, it invokes the learner profile management service to import data into the learner profile repository; the procedure call is forwarded through the network into a SOAP header to the matching method, and finally processed by the CIM Manager.

Figure 3. The management graphical interface

Moreover, these works are part of a project financed by the French national research agency, and aim at designing and implementing an intelligent tutor that should provide learners with helps according to their metacognitive profile. In order to gather this sub-profile, psychologist teachers have elaborated a quiz composed of thirty-seven questions determining the level of the learner for eight metacognitive attributes; for each question, learners had to select one response among six propositions. The core learner profile had to be extended, as all of these properties are not specified within the model; the management graphical interface was used to insert the missing attributes into the metacognitive sub-profile of the schema.

The next step consisted in building a web form matching with the quiz and integrating a client of the learner profile management service. When learners...
submit the form and after a computation on results, all values of the metacognitive attributes, together with the sex and the age of the learner, were respectively integrated into the metacognitive and identification learner sub-profiles.

The graphical interface shown on Figure 3 displays data provided by both Moodle and the web form. Presentation of data into the web page stimulates learners to study their metacognitive profile, as it is represented in a graphical manner. It thus helps learners to monitor their state of learning, their ability to plan strategies, adjust effort appropriately and evaluate the success of their on-going efforts to understand. As a matter of fact, learner profiles can also be delivered to any applications expressing a need for learner information as well.

4. Benefits and evaluation of the approach

4.1. Benefits

The lack of abstraction level and sharing mechanisms of existing learner profile standards, which prevent collaborative learning and promote enclosure, have spurred researchers to improve LIP-based systems: some operate on extensible profiles and offer query language to easily retrieve relevant information stored into learner profiles [13], whereas others exploit learner data to build profile-based applications like adaptive testing system [4][9]. Table 1 shows strengths and assets of the core learner profile compared to two existing systems and the LIP standard, and points out how our approach integrates features that we think are crucial to ensure sharing and reusing of learner profiles.

The core learner profile, based on IMS LIP, provides a high abstraction level and can be extended to satisfy additional pedagogical expectations. Moreover, the class TEL_ProfileCore allows modeling of profiles according to other user roles specific to TEL. For instance, one could extend the core profile by adding class(es) composed of specific attributes describing tutor profiles.

The open learner profile is part of the whole CIM schema, and various native classes already specify information specific to users. For instance, the CIM Device model details all devices available to users; it is thus possible to discover the keyboard layout, the screen resolution, or if the user owns a Tablet PC or a scanner. The CIM System model describes, among other information, the name, type and version of the operating system and file system. All these data represent a powerful and trusted source of information when systems come to provide personalized learning material.

<table>
<thead>
<tr>
<th></th>
<th>IMS LIP</th>
<th>REPro</th>
<th>CosyQTI</th>
<th>Core learner profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard integration</td>
<td>Yes (PAPI)</td>
<td>Yes (LIP)</td>
<td>Yes (LIP)</td>
<td>Yes (LIP + CIM)</td>
</tr>
<tr>
<td>Extensibility</td>
<td>No</td>
<td>Open</td>
<td>No</td>
<td>High abstraction level</td>
</tr>
<tr>
<td>Support architecture</td>
<td>None</td>
<td>None</td>
<td>Specific</td>
<td>De facto Standard (WBEM)</td>
</tr>
<tr>
<td>Query language</td>
<td>No</td>
<td>PMDL</td>
<td>No</td>
<td>WQL or CQL</td>
</tr>
<tr>
<td>Functional status</td>
<td>Specification</td>
<td>Prototype</td>
<td>Ready to use</td>
<td>Ready to use</td>
</tr>
</tbody>
</table>

The concrete use cases demonstrate how the framework makes it possible for disparate and heterogeneous applications to transparently collaborate in order to bring complementary information about a common learner profile; data automatically gathered from Moodle combined with those resulting from the quiz show that main attributes of the core learner profile can easily be gathered, thus significantly increasing the amount of data available to describe a learner. The metacognitive questionnaire also validates the generic and extensible capabilities of our approach, demonstrating that any specific properties can be defined according to precise expectations.

The management graphical application shows that learner profiles are easily retrieved by end-users and client applications in order to be reused, thanks to the CIM query language. Teachers can compare evolution of learner profiles and thus enhance their learning scenario, or identify learners that need some prerequisites, whereas learners can reflect on their own profile.

4.2. Evaluation

The model-driven approach explicitly defines data to collect through a predefined model, and may thus prevent considering unpredicted data that are useful in term of pedagogy. However, the high abstraction level of our model, combined with the profile management service, tackle this issue.

Several works interested in tracking users are based on multi-agent systems. Some studies shown that such environments are harder to deploy and monitor than centralized ones, and that they imply a significant
usage of CPU resources. The distributed and decentralized framework aims at filling these lacks: tracking components are strongly independent thanks to a 3 tiers architecture, and processes executed on the client side are restricted to the collect of data.

Finally, the CQL language makes it easy to interact with the model. Since it is based on the ISO SQL and W3C XML-Query languages, complex queries can easily be elaborated to retrieve data at various granularity levels.

5. Conclusions and further works

We presented in this paper a model-driven approach to share and reuse learner profiles between heterogeneous environments. The core learner profile integrates the LIP standard together with some metacognitive properties, while some other classes have a strong focus on extensibility and genericity.

This framework is based on the WBEM standardized architecture, which should encourage adoption of our proposal. Indeed, ubiquitous operating systems natively implement a WBEM tool to facilitate local and remote management of hosted systems and applications; since our open learner profile can be integrated into any WBEM-compliant tools, information specific to learners can be handled by any WBEM component. In addition, two management services located on top of WBEM provide web-based applications with very simple interfaces to either modify and extend the learner schema, or extract data from and insert data into the learner profile repository. These two distinct layers thus facilitate sharing and reusing of learner profiles between heterogeneous systems and environments, as management services can federate and merge information stored into distributed WBEM tools.

A short-term perspective consists in elaborating a component to extract data from the Microsoft™ operating system through its native WBEM components, and to externalize information to the learner profile repository. A huge amount of data is available, and should bring a significant value to our framework. Indeed, various information about the software and hardware environments of the learner will automatically be discovered and very helpful to personalize learning resources, training and systems.

6. References


