A Framework for Browsing, Manipulating and Maintaining Interoperable Learner Profiles*

Peter Dolog and Michael Schäfer

L3S Research Center, University of Hannover,
Expo Plaza 1, 30539 Hannover, Germany,
dolog@l3s.de

Abstract. Learners are assessed by several systems during their life-long learning. Those systems can maintain fragments of information about a learner derived from his learning performance and/or assessment in that particular system. Customization services would perform better if they would be able to exchange as many relevant fragments of information about the learner as possible. This paper presents the conceptualization and implementation of a framework which provides a common base for the exchange of learner profiles between several sources. The exchange representation of learner profiles is based on standards. An API is designed and implemented to create/export and manipulate such learner profiles. The API is implemented for two cases, as a Java API and as web services with synchronized model exchange between multiple sources. Application cases of the API are discussed shortly as well.

1 Introduction

Each user adapted service or application needs a user profile to perform the adaptation accordingly. In the area of education, several approaches have been proposed to collect information about users such as preferences, following clicking behavior to collect likes and dislikes, and questionnaires asking for specific information to assess learner features (e.g. tests, learner assessment dialogs, and preference forms). In addition, several tools have been designed to improve learner models by open active learner modelling. The variety of use cases are supported by such tools like maintaining and comparing the student's own and the system's believes about his knowledge [3], multiple choice questionnaires [2], collaborative peer assessment in discussions [**?**], and dialogues with interactive topic maps [4].

These systems can be seen as services to improve user or learner models in open environments. Different users may prefer a different style of evaluation and thus may

* This work is partially supported by EU/IST ELENA project IST-2001-37264.

want to choose one or more of them which are the most suitable for them to evaluate their profiles. To benefit from such heterogeneous services, an interoperable learner profile and an infrastructure to support its exchange should be provided. The following questions arise: *how to represent the learner profile*, *how to access the learner profile*, and *how to provide an extensible API to process heterogeneous profiles*.

The rest of the paper is structured as follows: Section 2 discusses standard based representations of learner profiles, its instantiation, and mappings from internal data models. Section 3 discusses how the models can be accessed by means of a Java API, webs services, querying infrastructure for RDF, and application cases which have been implemented. Section 4 provides a summary and an outline of possible further work.

2 Learner Profile Exchange Model

In order to be able to exchange a learner profile between e-Learning and learner assessment systems, we need to provide explicit information about what is going to be exchanged, which values of the specific subject are considered and how the information is bound to a learner. Learner profile standards and open specifications provide us with a representation for subjects of exchange, e.g. learner performance, portfolio, preferences, learning style, certificates, evaluations, and assessment. Domain ontologies provide us with exchangeable/sharable models of domains. Such ontologies can model either the domain which will be overlaid in the learner profile, learner competencies/skills, or can model stereotype structures.

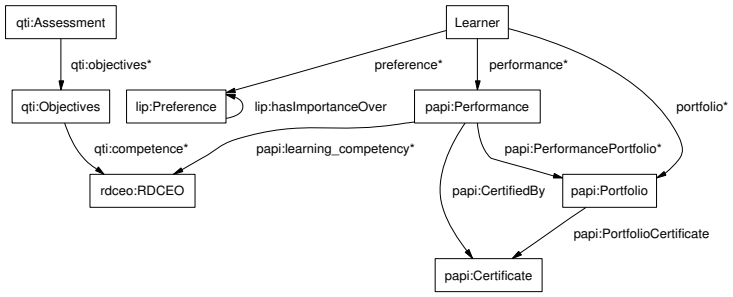

Fig. 1. An excerpt of a conceptual model for learner profile based on standards

Learner Ontology. Figure 1 depicts an excerpt of a learner profile ontology configured from fragments based on three specifications[1]. The conceptual model describes a situation where a learning performance[2] of a student is exchanged as his achieved competency[3][4] records. The competencies have been evaluated by learner assessment (e.g.

[1] Refer to http://www.l3s.de/ dolog/learnerrdfbindings/ for an extended model of the learner profile.

[2] IEEE PAPI is being used to model performance and portfolio: http://ltsc.ieee.org/archive/harvested-2003-10/working_groups/wg2.zip.

[3] IMS reusable definition of competency and educational objectives (IMS RDCEO).

[4] Refer to http://www.imsglobal.org/ for all IMS specifications.

tests) and were derived from learning objectives of tests[5]. Furthermore, all other educational activities, further materials, and projects created within the activities are reported within the portfolio of the performance. Additional information which is reported under preferences[6] comprises language, device, resource and learning style preferences. The standards and open specifications guarantee wider acceptance between eLearning systems and as such can be seen as good candidates for the learner exchange models.

Instantiation and Mappings from Internal Models. The tools, which use a different internal data model and would like to participate in an exchange of learner profiles, have to provide mappings between their internal data model and the exchange model. Besides that, an evidence about how a learner model was derived should be provided to allow other systems to interpret the model correctly. If we take for example an overlay model of a domain, the sub domain concepts are bound to the learner performance together with time stamps, certificates and resources which contributed to the performance. The sub concepts, referenced as competency hierarchies, are further bound to assessment resources like dialogs used, questionnaires filled in with their results, activities with concept maps performed, and so on. This information allows to trace back the computation of particular learner model fragments and to determine how they contribute to the overall integrated model.

3 Accessing the Learner Profile

Figure 2 depicts several scenarios of how to access and exchange learner profile fragments. The fragments can be accessed programmatically by the use of a Java API, the web service which exports the learner model through the API and acts as a learner model server, and through a query infrastructure for RDF repositories like Edutella [9].

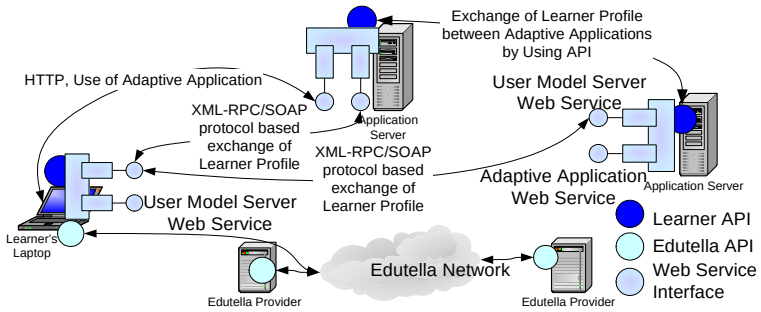

Fig. 2. The use of the API in several scenarios

Access through Java API. We build a Java API which is structured according to the learner profile fragments mentioned above. The API is meant to be used to retrieve, insert, and update the learner profiles stored in the structures described above. The API

[5] IMS questions and test interoperability (IMS QTI).
[6] IMS learner information package (IMS LIP).

defines a class and properties for each class from the RDFS for the learner model. The interface provides access functions for getting, deleting and updating a model of the fragment. It provides further functions to derive additional information or to process more complex manipulations over referenced information types as well. The API is implemented for the RDF representation (instances of the RDFS described above). The API is easily extensible by providing further specializations if additional extensions and interface implementations for local repositories and data models are needed.

Access through API as Web Services. The second implementation is provided through web services where several clients can access one model which is persistent on one server. The server holds the main model, i.e. the data of a learner profile gathered from several sources, and handles all requests from the clients. Each client is uniquely identified at the server and can be used by a browsing or assessment system. Furthermore, a client can be used by other learning systems which want to make use of the learner profiles or which want to contribute to them. The model can be accessed directly by invoking functions of a web service or in a synchronized replicated way; i.e. each client has its own repository which is synchronized with the main server every time a change occurs. The web services framework can be used in a distributed way as well (several servers exchanging learner models between each other).

Retrieval through RDF querying infrastructure. The learner profiles are created in RDF. Therefore, a query infrastructure for RDF data is another access option. Edutella provides a datalog-based language to query RDF data provided in a distributed P2P environment. This option enables to collect various fragments by utilizing for example the algorithm from [5]. Another advantage of the P2P sharing infrastructure used with the learner profiles is that it can facilitate an expert finding based on the provided profile which can be queried by people who need a help in learning.

Recent Application Cases of the Framework. The API has been tested at a simple browsing and dialog system (Learner Browser) and with the UML- guide system [7]. In the UML-Guide the API is used to record clicking behavior of the learner in a knowledge map by means of events triggered when a particular knowledge map item is clicked. In the Learner Browser, the profile can be browsed through several categories of a learner data with possibility to use it for self-reflection; i.e. to update simple categories like preferences, add a competence based on an evaluation by a test, and so on. Further implementations towards other assessment services are envisaged.

4 Conclusions and Further Work

We have described a framework which utilizes standards to make learner profiles interoperable. A user model server similar to the one described in [8] is implemented by making use of the framework. The server is accessible as a web service. A Java API was implemented making use of the framework to allow other systems to plug into the standard based learner modelling component. We have also discussed how to map internal data models of user modelling systems to the standard based descriptions to enable exchange of learner models.

In our further work we would like to further investigate how this API can be used within P2P environments similarly to [10]. We have made first steps towards such an environment in [6, 5] where we discussed how to collect fragments. The API provides us with manipulation functionalities. The combination of both might lead to interesting solutions. Privacy will be further investigated as well.

References

1. Bull, S., Brna, P., Critchley, S., Davie, K., and Holzherr, C.: The missing peer, artificial peers and the enhancement of human-human collaborative student modelling. In *Proc. of Intl. Conference on Artificial Intelligence in Education*, 269–276. IOS Press, 1999.
2. Bull, S. and Nghiem, T.: Helping learners to understand themselves with a learner model open to students, peers and instructors. In Paul Brna and Vania Dimitrova, editors, *Proc. of Workshop on Individual and Group Modelling Methods that Help Learners Understand Themselves, Intl. Conference on Intelligent Tutoring Systems*, 5–13, 2002.
3. Bull, S. and Pain, H.: Did I say what I think I said, and do you agree with me? In *Proceedings World Conference on Artificial Intelligence in Education, AACE*, 501–508, 2002.
4. Dimitrova, V.: Style-olm interactive open learner modelling. *Intl. Journal of Artificial Intelligence in Education*, 13:35–78, 2003.
5. Dolog, P.: Identifying relevant fragments of learner profile on the semantic web. In *Proc. of SWEL'2004 — Intl. Workshop on Semantic Web for eLearning, Intl. Semantic Web Conference 2004*, Hiroshima, Japan, November 2004.
6. Dolog, P. and Nejdl, W.: Challenges and benefits of the semantic web for user modelling. In *Proc. of AH2003 — Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, WWW2003 Conference, Hypertext'03 Conference, User Modelling Conference 2003*, Budapest, Hungary, Nottingham, UK, Johnstown, PA, USA, 2003.
7. Dolog, P. and Nejdl, W.: Using UML and XMI for generating adaptive navigation sequences in web-based systems. In Perdita Stevens, Jon Whittle, and Grady Booch, editors, *Proc. of UML 2003 — The Unified Modeling Language. Model Languages and Applications. 6th Intl. Conference*, volume 2863 of *LNCS*, 205–219, San Francisco, CA, USA, October 2003. Springer.
8. Fink, J. and Kobsa, A.: User modeling in personalized city tours. *Artificial Intelligence Review*, 18(1):33–74, 2002.
9. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., and Risch, T.: EDUTELLA: a P2P Networking Infrastructure based on RDF. In *In Proc. of 11th World Wide Web Conference*, 604–615, Hawaii, USA, May 2002.
10. Vassileva, J., McCalla, G., and Greer, J.: Multi-agent multi-user modelling in I-Help. *User Modeling and User-Adapted Interaction*, 13(1–2):179–210, 2003.